Linked Data for Software Security Concepts and Vulnerability Descriptions

by

Arnav Prabodh Joshi

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Masters in Computer Science
2013

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**
4. TITLE AND SUBTITLE **Linked Data for Software Security Concepts and Vulnerability Descriptions**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Maryland, Baltimore County,Computer Science and Electrical Engineering,Baltimore,MD,21250**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S)
		11. SPONSOR/MONITOR’S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

The Web is often the first source of information to track software vulnerabilities, exploits and cyber attacks. An important source is information found in text from security bulletins, vulnerability databases, news reports, cybersecurity blogs and Internet chat rooms. However, the data representation and interpretation of such unstructured text pose certain limitations on the automation of vulnerability management, and obtaining further contextual information from other related resources. We present an automatic framework that generates and publishes a RDF linked data resource for software security concepts and vulnerability descriptions. Vulnerability descriptions from the National Vulnerability Database (NVD) are aligned with concepts from parallel repositories such as the Common Weakness Enumeration and Common Platform Enumeration. These concepts are represented in RDF using relevant concepts from a custom ontology that models the relationships between classes and entities for the cybersecurity domain. The unstructured sources of information from the NVD are then mapped to related concepts from DBpedia using object properties from the ontology. This system leverages paradigms of the Semantic Web to effectively process unstructured text into a rich resource of machine-understandable information. The RDF linked cybersecurity data collection will make it possible for applications to look up metadata and facilitate searching. Our results demonstrate an effective model for linking key security concepts to relevant resources on the Web. We outline the use of Linked Data technologies to facilitate consumption of information related to security exploits that can be further used for vulnerability identification, mitigation and prevention efforts.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT **Same as Report (SAR)**	18. NUMBER OF PAGES **78**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

ABSTRACT

Title of Thesis: Linked Data for Software Security Concepts and Vulnerability Descriptions

Arnav Prabodh Joshi, Masters in Computer Science, July 2013

Thesis directed by: Dr. Timothy W. Finin, Professor
Department of Computer Science and
Electrical Engineering

The Web is often the first source of information to track software vulnerabilities, exploits and cyber attacks. An important source is information found in text from security bulletins, vulnerability databases, news reports, cybersecurity blogs and Internet chat rooms. However, the data representation and interpretation of such unstructured text pose certain limitations on the automation of vulnerability management, and obtaining further contextual information from other related resources. We present an automatic framework, that generates and publishes a RDF linked data resource for software security concepts and vulnerability descriptions.

Vulnerability descriptions from the National Vulnerability Database (NVD) are aligned with concepts from parallel repositories such as the Common Weakness Enumeration and Common Platform Enumeration. These concepts are represented in RDF using relevant concepts from a custom ontology that models the relationships between classes and entities for the cybersecurity domain. The unstructured sources of information from the NVD are then mapped to related concepts from DBpedia using object properties from the ontology. This system leverages paradigms of the Semantic Web to effectively process unstructured text into a rich resource of machine-understandable information. The RDF linked cybersecurity data collection will make it possible for applications to look up metadata and facilitate searching. Our results demonstrate an effective model for linking key

security concepts to relevant resources on the Web. We outline the use of Linked Data technologies to facilitate consumption of information related to security exploits that can be further used for vulnerability identification, mitigation and prevention efforts.

Dedicated to Aai, Baba and Maitha, my source of inspiration, my pillars of strength; to Neha, my support system; and my teachers, mentors and friends.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude towards my advisor Dr. Tim Finin for his kind supervision, productive discussions and suggestions that guided me in my research. He has always encouraged me to explore new concepts and pursue newer research problems. I credit my thesis contribution to him, and many skills that I have gained while working with him. My special thanks go to Dr. Anupam Joshi for all of his valuable inputs throughout my work at the Ebiquity research lab. His keen observations and insight were very helpful throughout the research process. Collectively, I would also like to thank my thesis committee members Dr. Tim Finin, Dr. Anupam Joshi and Dr. Tim Oates for their time, suggestions, and for graciously agreeing to be on my committee, and always making themselves available. All the Ebiquity group members have been extremely supportive in building an exciting atmosphere conducive to research. It has been a great joy working with them throughout the year. I extend my sincere thanks to the National Science Foundation, and the Air Force Office of Scientific Research for funding this research. Lastly, I would like to thank my friends back home in Pune, and those in Baltimore, who made me feel closer to home, and supported me throughout my Masters degree. I cannot thank them enough.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Cybersecurity is a critical concern as society has become increasingly interconnected, and reliant on, a global system of computers, communication networks and software systems. Cyber crime has continued to become more professional, with the emergence of increasingly powerful methods of intrusion and exploits. For example, cyber criminals targeted users of Skype, Facebook and Windows using multiple blackhole exploits in October 2012 (1). In the current state, systems are reported to be under threat from vulnerabilities that are already published. One of the major reasons is that these systems are not patched on a regular basis, which can be attributed to user apathy, and insufficient security control (lack of intrusion detection hardware/software). Information regarding these patches is available and has a rich semantic structure which is clear to a human reader. However, such sources remain mostly unavailable to and unused by automated intrusion detection systems.

There are several sources for sharing information on security that give guidance and describe mitigation for publicly disclosed vulnerabilities. These security advisory sources comprise of both government-curated repositories and technical blogs, security bulletins and reports by companies for their range of software products. These repositories address security changes and threat trends that might affect a system's overall security, such as “defense in depth” security enhancements or changes that are unrelated to security vulnerabilities. Vulnerability enumeration schemes such as CVE[1], CWE[2], CVSS[3] and NVD[4] list vulnerabilities and exposures, provide common names and identifiers for publicly known problems, a consolidated list of weaknesses in software that can lead to exploitable security vulnerabilities, and metrics to measure the severity of a threat. Although the above mentioned collections of data provide a structured representation, they constitute as raw data which needs to be processed for automated consumption.

Additionally, many concepts, even in such detailed descriptions, remain hidden in unstructured text; such as the systems that are likely to be affected, the operating systems environment for which the attack can occur, the versions of products that are vulnerable, and the relationships between these entities. Representing the cybersecurity-related concepts described in these extensive repositories in the form of a structured, semantically-rich, machine-understandable format will allow automation of the data consumption, enable gathering information described in other documents and resources linked to these security concepts, and thereby help enrich the corpus of machine-readable data with information from heterogeneous data sources.

Vulnerabilities are also mentioned in security bulletins and blogs, which typically are narrative descriptions that include the above mentioned relationships. Collaborating and expressing these sources of information in a structured, semantic, machine-understandable format can allow prevention of possible “zero-day” attacks. The enumeration schemes and the advisory sources can enable data sharing across separate vulnerability capabilities, when represented in a semantically rich and structured format. The Linked Data principles explain how to identify, access, describe and interlink data on the Web using existing Web

[1] Common Vulnerabilities and Exposures - http://cve.mitre.org/

[2] http://cwe.mitre.org/

[3] http://www.first.org/cvss

[4] http://nvd.nist.gov/

standards, such as the HyperText Transfer Protocol (HTTP), Uniform Resource Identifier (URI) and the Resource Description Framework (RDF). To be able to capture semantics of data, Linked Data extends the scope of the traditional Web from documents to encompass concepts of the real world which make up the data. We believe that extending these principles to the corpus of cybersecurity-related documents, can allow machines to process and understand the entities, and hence speedup information exchange from data across different organizations.

We describe a linked data generation framework that extracts cybersecurity-relevant entities, terms and concepts from unstructured text from NVD. These extracted concepts are then mapped to relevant linked resources on the Web using an OWL ontology and represented as RDF linked open data. The immediate external benefit of a linked data resource is that organizations will be able to link their data with similar or related datasets. This can help improve visibility and use of data that otherwise would be stuck in data silos. Sharing data also can help improve the overall quality of data as well as make it more useful to end-users - such as system administrators and security practitioners. We believe that such a publicly available linked open data resource will help organizations uncover knowledge from multiple sources of cybersecurity-related data on the Web, and thereby permit the use of a system to automatically ingest and reason over this data, and help intrusion detection and prevention in real time.

Chapter 2

BACKGROUND AND RELATED WORK

In this chapter we discuss the motivation behind generating RDF linked cybersecurity data and some related work that forms the basis of this research.

2.1 Sources for Sharing Information Security

Several repositories and security advisory sources address security changes and threat trends that might affect the overall security of a computer system. These sources can be used in a variety of ways to enhance the process of detection of an attack.

2.1.1 The National Vulnerability Database (NVD)

The National Vulnerability Database (NVD) is the U.S. government repository of standards based vulnerability management data, represented using the Security Content Automation Protocol (SCAP) (2). The main purpose of NVD is to provide previously unavailable technical capabilities and offer support for different vulnerability standards. This is attained via integration of all publicly known and available vulnerability resources. As of June 2013, there are over 56814 CVE vulnerabilities, 246 US- CERT[1] alerts, and 8140

[1] http://www.us-cert.gov/

OVAL[2] queries registered with the NVD collection. The publication rate is approximately 14 vulnerabilities a day.

All NVD data is freely available in the form of XML feeds for vulnerabilities that are published in a particular year. NVD uses the Common Vulnerability Scoring System (CVSS), an open standard for assigning vulnerability impacts, that is also used by a variety of organizations. NVD is the only repository of CVSS scores for all CVE vulnerabilities. Since 2002, the NVD repository has been published at regular intervals. The NVD datasets are updated immediately with raw information whenever a new vulnerability is reported to the CVE repository, and iterated to a valid, confirmed source after analysis[3]. The raw vulnerabilities are analyzed by NVD analysts and augmented with vulnerability attributes (e.g. vulnerable product versions and operating environments).

NVD assigns vulnerabilities the following impact types: confidentiality ("allows unauthorized disclosure of information"), integrity ("allows unauthorized modification"), availability ("allows disruption of service"), and security protection ("provides unauthorized access"). The "provides unauthorized access" category refers to getting some sort of general privileges in the application or entire computer (e.g., getting root access or an application account). This category has three possible sub-categorizations: user level access to the operating system, administrator privileges, and any other type of privileged access. NVD only records what impact types a vulnerability directly allows. Several attacks allow an attacker certain general privileges on a computer or within an application (e.g., the ability to execute code). By exploiting this privilege, an attacker can violate confidentiality, integrity, and availability for a system. However, the NVD schema does not cover this aspect since some vulnerabilities (usually buffer overflows) allow both direct violation of confidentiality, integrity, or availability (usually availability) and then also allow one to

[2] http://oval.mitre.org/

[3] http://nvd.nist.gov/faq.cfm

gain general “unauthorized access”.

2.1.2 Common Weaknesses Enumeration (CWE)

Software weaknesses can be defined as the “flaws, faults, bugs, vulnerabilities, and other errors in software implementation, code, design, or architecture, that if left unaddressed could result in systems and networks being vulnerable to attack”[4].The CWE dictionary has evolved based on previous analysis of attacks that happened with a similar signature on a particular class of software products.

NVD analysts score CVEs using CWEs from different hierarchical levels. This allows analysts to score CVEs at a fine or coarse granularity, which becomes vital since specificity of different CVEs varies at certain levels. CWE provides valuable information, essentially a detailed summary of the weakness, the base metrics to calculate the severity score for a vulnerability, consequences description, and possible mitigation information. Modeling this data with the corresponding NVD repository, will help integrate information related to a particular vulnerability. This will essentially help to start an initiative to obstruct vulnerabilities at the root cause, by educating software professionals, designers, architects, and programmers on how to eliminate the most common threats that can target a software product or the overall computer system.

2.2 Motivation

Information sources such as the NVD and IBM XFORCE[5] provide XML feeds that report vulnerabilities with varying degrees of detail. To the best of our knowledge, these repositories consolidate information present across multiple data sources, though are manually monitored. These dictionaries not only contain redundant or overlapping information,

[4] http://cwe.mitre.org/about/faq.html

[5] http://xforce.iss.net/

but also miss out on important concepts such as the means and consequence associated with an attack and the version of a software product. In part, similar information is available in cybersecurity blogs such as Krebsonsecurity[6] and the CERT blog[7]. However, it is completely unstructured, which can lead to an information overload, especially during threat analysis of a system. Furthermore, analyzing and integrating multiple textual resources can become a cumbersome task for system administrators. Expressing these informal sources as a linked RDF knowledge base will not only enhance distribution of security information, but also the discoverability of security-related concepts. Information describing cybersecurity terms when converted to RDF formatted data, can be useful for semantic analysis of vulnerabilities, and to avail statistics on latest threat trends based on a per-product basis.

Linking heterogeneous sets of documents to the cybersecurity linked data collection can add more context for a vulnerability description. For example, a CVE description for a vulnerability caused by a buffer overflow attack on Adobe Acrobat, contains limited information on the nature of the attack, or the possible consequences it generates. Linking the CVE resource to linked documents describing “Buffer Overflow” and identifying it as a means of an attack; and then based on the available information, infer that a possible denial of service can occur for the software product provides a concrete description of the software threat. Hence, searching for information about possible threats that are targeted towards a software product can be expedited, and would not be limited to semi-structured data model that NVD offers. This semantic search capability can be leveraged via querying a RDF linked cybersecurity data corpus, and can be consumed by an application such as a situation-aware intrusion detection system.

[6] http://krebsonsecurity.com/

[7] http://www.cert.org/

2.3 Related Work

An example of the above mentioned system was demonstrated in part by More et al. (3). The work demonstrates an effective reasoning over such a semantically rich data for a situation aware intrusion detection system. The framework requires a condensed source of web resources that provide meaningful information about the threat, and data sources that provide entities that map well into the ontology. Our approach provides automation to generate such a linked data resource that consolidates both structured and unstructured sources of data, and gives adequate insight on the nature of the attack, thereby allowing effective intrusion detection and prevention.

Mulwad et al. (4) describe a preliminary prototype that analyzes relevant text snippets from the Web and generates assertions about vulnerabilities, attacks and threats. The system extracted concepts of interest and queried Wikitology, a knowledge base of entities from DBpedia[8], Yago[9], Freebase[10] and similar sources. The classification mechanism and the spotted concepts were limited to the identification of two classes: the *means* and the *consequence* of an attack. We adopted an approach that uses a Conditional Random Field (CRF) algorithm trained with ground truth annotations (5) to identify and classify mentions of entities and concepts that goes beyond their simple approach in terms of precision and recall.

The quality of the extracted concepts from free text largely depends on the method applied for concept spotting. More et al. (3) used OpenCalais (6), an open-source named entity recognizer that is trained to identify entities based on classes in Wikipedia. OpenCalais was designed to recognize general entities such as people, places and organizations and could not identify many of the entities and concepts related to cybersecurity. Similar

[8] http://dbpedia.org
[9] http://www.mpi-inf.mpg.de/yago-naga/yago/
[10] http://www.freebase.com/

experiments were run on the NERD information extraction framework (7), which failed to identify relevant technical jargon from the given piece of security-related text. These annotation tools are designed to capture information based on a custom ontology that models people, places and organizations. The standard Stanford Named Entity Recognition (NER) (8), without proper feature filtering, also does not identify key cybersecurity concepts. Our approach used a cybersecurity entity and concept spotter developed by Lal (9), that was primarily trained to identify entities (e.g., software products and operating systems) and concepts (e.g., denial of service and buffer overflow) which are related to computer security, threats and vulnerabilities in software products.

Khadilkar et al. (10) demonstrated the concept of a semantic model to facilitate information representation and described an ontology for the National Vulnerability Database. The ontology models information for software products and generic security concepts, though is unable to characterize and capture information from unstructured sources of information. The vocabulary was geared towards use cases of applicability between different IT products. Their approach described an identification strategy for specific products and was limited to a few vendors. Further, the information modeled in the ontology was limited to data represented as strings in the NVD corpus, and not linked to any relevant entity on the Web. To the best of our knowledge, there are no previous efforts taken towards generating linked data from security concepts identified from text.

Undercoffer et al. (11) specify an ontological model for categorizing computer attacks. The ontology used taxonomic characteristics of an intrusion to be limited to specific classes and attributes that are centered on the target of an attack. The ontology modeled for identifying an attack on a particular system component, and broadly classified to the system, process and network levels. Our framework consolidates information across different knowledge bases and carries out concept-spotting for entities of interest, that can initiate characterization and understanding of the overall nature of the attack.

2.4 Semantic Web and Linked Data

Creating and publishing data following Linked Data principles helps search engines and humans to find, access and re-use data. Once information is found, computer programs can re-use data without the need for custom scripts to manipulate the content. Linked Data refers to an incremental framework for deploying data, by hyperlinking machine-readable data sets to each other using Semantic Web techniques, especially via the use of RDF and URIs. Publishing data on the Semantic Web with machine interpretable representations facilitates more structured and efficient access to the resources; however semantic descriptions, without being linked to other existing data on the Web, would be mostly processed locally and based on the domain descriptions (i.e. ontologies) and their properties. Linking data to other resources on the Web enables obtaining more information across different domains. Publishing annotated and interconnected data is the underlying principle of the Web of Data (12). Berners-Lee (12) discusses the four basic principles for linked data as

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URIs so that they can discover more things.

Linked Data (13) enables publishing structured, machine-readable interpretation of heterogeneous sources of information. As defined by Bizer et al. (14), it is “a set of best practices for publishing and connecting structured data on the Web.” It focuses on interconnecting data and resources on the Web by defining relations between ontologies, schemas and/or directly linking the published data to other existing resource on the Web.

For example, an NVD entry might contain the following real-world concepts: the *vulnerability*, the affected *product*, the *weakness* class the attack belongs to, and associated mitigation information. Each concept is then described in the form of simple sentences, e.g., “A buffer overflow vulnerability with CVE ID *CVE-2013-0610* affects product Adobe Acrobat 9.3 and above”, in such a way that machines comprehend the meaning of these security concepts. The vulnerability description, its relationship with the software product, and the actual dereferenceable resource describing the security concept is defined in separate knowledge bases. As opposed to the hypertext links that connect documents on the Web, such link not only connects two datasets together, but defines the semantics of the connection, e.g., from the link we know that one piece of data “affects the product” described by the other piece of data.

Over the past four years, a huge collection of Linked Data from different domains emerged on the Web, including media, government, geographic, life-sciences and library data. This corpus of diverse semantically rich machine-readable data is called the Semantic Web or a Web of Data. The success of Linked Data is evident by the increasing amount of data published directly by organizations such as the BBC, in its internal content production systems (15).

In our work, we leverage the concepts of the Semantic Web and Linked Data to the cybersecurity domain by building an RDF data store for vulnerabilities, severity metrics, affected products, sources on the Web that identified and further describe the full nature of the attack, and any remedial information if available. Relevant information about these concepts from other sources on the Web can be interlinked to give a denser graph of linked entities. With NVD data represented as RDF linked data, the task of finding all vulnerabilities pertaining to a single product version is reduced to the task of traversing the product-vulnerability dependency graph.

Additional contextual information obtained through establishing meaningful semantic links can help consolidate available information regarding a security threat. Moreover, the data representation for this interlinking will be in a structured, machine-readable format enabling faster, automated data consumption. The linked data resource can help improve the discoverability of data through the use of SPARQL[11] queries, SPARQL endpoints and resolvable URIs. The advent of SPARQL 1.1, which allows matching entities based on a regular expression, enhances the search capability over the linked cybersecurity data graph to retrieve collections of vulnerabilities. It also helps in use cases such as distinguishing relevant vulnerabilities based on a product term or version. Such an interlinked corpus of data will enable stakeholders to share security-related information in a single resource, create business intelligence, support automated decision making systems and thereby speedup the exchange and digestion of information across different organizations.

2.5 Definitions

XML and RDF prefixes In RDF, prefixes are used to identify the specific vocabulary that a particular class, object property or data property is associated with. Table2.1 lists the prefixes and associated schemas or ontologies used in this study.

[11] http://www.w3.org/TR/sparql11-overview/

Table 2.1. XML and RDF schemas and ontologies used in this study

XML Prefix	XML Schema
xmlns	http://scap.nist.gov/schema/feed/vulnerability/2.0
xmlns:vuln	http://scap.nist.gov/schema/vulnerability/0.4
xmlns:cvss	http://scap.nist.gov/schema/cvss-v2/0.2

RDF Prefix	RDF Schema
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
dbpedia-owl	http://dbpedia.org/ontology/
ebqids	http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/IDSOntology.owl#

Chapter 3

SYSTEM ARCHITECTURE

In this chapter we present our proposal for publishing Linked Cybersecurity Data. We start by introducing the underlying methodology in Section3.2. The rest of the work is divided according to the two main successive steps of the methodology: data modeling and data publication.

3.1 Overview

Figure 3.1 shows the organization of the linked data generation framework, which can be divided into the following components.

1. Data Modelling

   (a) A cybersecurity-specific vocabulary that models entities and elicits concepts described in the data corpus.

2. Data Publication

   (a) An RDF triple generator that generates triples for the vulnerability resources, which are aligned to the ontology. The information modeled in a vulnerability resource is based on the CVE, CWE, NVD schema and other sources such as security bulletins, technical reports among others.

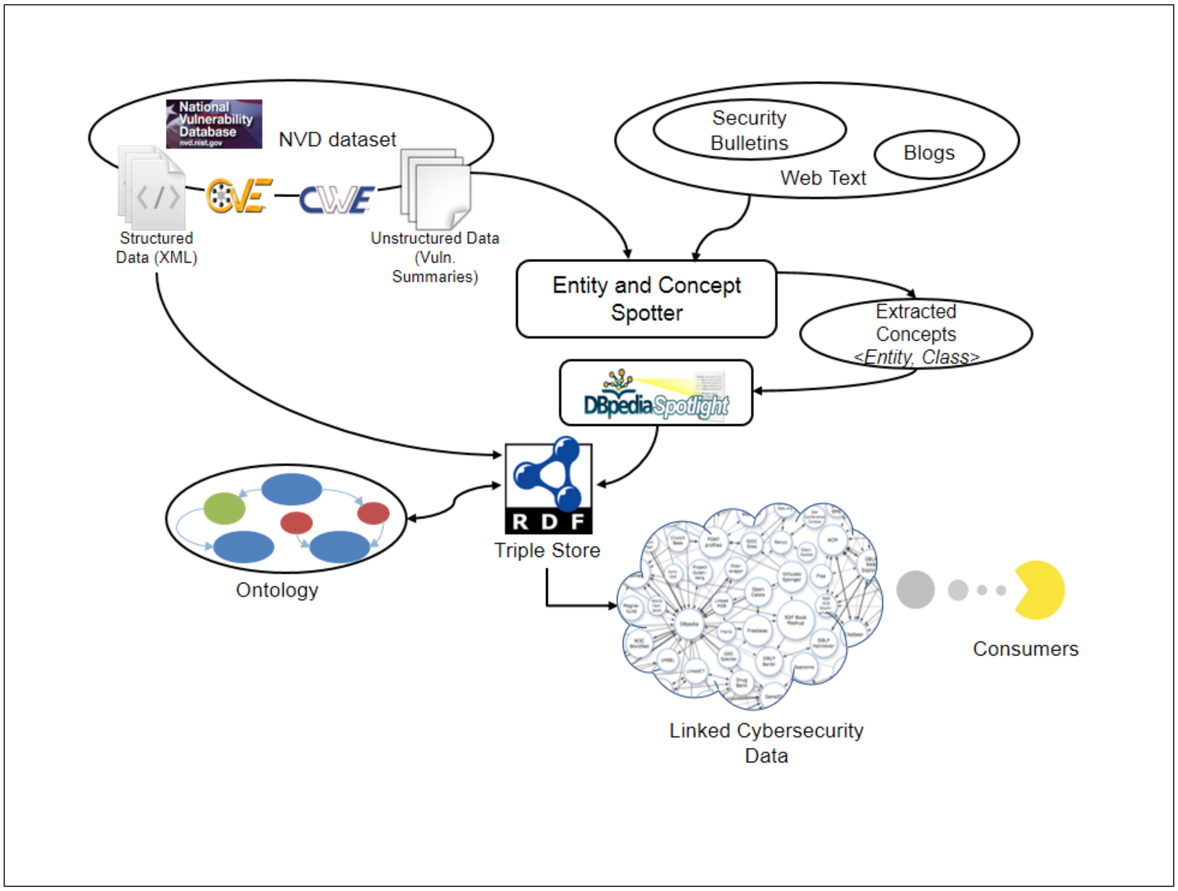

National Vulnerability Database
nvd.nist.gov
NVD dataset
CVE
CWE
Structured Data (XML)
Unstructured Data (Vuln. Summaries)
Security Bulletins
Blogs
Web Text
Entity and Concept Spotter
Extracted Concepts <Entity, Class>
DBpedia Spotlight
RDF
Triple Store
Ontology
Linked Cybersecurity Data
Consumers

FIG. 3.1. System architecture for extracting linked cybersecurity data from text

(b) Establishing links for the concepts extracted from the unstructured pieces of text and mapping them back to the vulnerability resource.

(c) Deploying linked cybersecurity data

In the following sections, these components will be described in detail.

3.2 The IDS Ontology

Vocabularies play a very important role in Linked Data, specifically to help with data integration. An ontology is “a formal model that allows knowledge to be represented for a specific domain" (16). An ontology describes the types of things that exist (classes), the relationships between them (properties) and the logical ways those classes and properties can be used together (axioms).

The modeling process in the context of RDF refers to capturing the context of data and define the relationships of the data. By doing so, we create a setting for generation of high quality Linked Data, since capturing information in a well defined context ensures better understanding, proper reuse, and is critical when establishing linkages to other data sets.

We developed the IDS ontology by manually aligning the different classes using their definitions and providing a best coverage of the principal axioms described in the NVD schema.During the IDS ontology modeling step, we considered the key design considerations that need to be covered to prepare and publish the NVD data corpus as linked cybersecurity data on the Web. These considerations break down into three areas, each of which maps onto one or two of the Linked Data principles (described in Section 2.4) -

- Identifying and eliciting security-related concepts
- Mapping these concepts to dereferenceable resources with HTTP URIs that describe the concept in detail.
- Describing the object and data properties associated with each concept in OWL[1].

3.3 Modeling Cybersecurity Concepts

We modeled the IDS ontology[2], partially depicted in Figure 3.2, to represent concepts and entities that are relevant to the cybersecurity domain. We used the Protege Ontology

[1] http://www.w3.org/2004/OWL/

[2] http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/IDSOntology.owl#

Editor to facilitate the modeling of concepts from the data into an ontology (17), using the OWL 2 Web Ontology Language (18). This vocabulary was originally developed by Undercoffer et al. (11), further enhanced by More et al., and is expected to continue to evolve to cover additional concepts. We extended the ontology to provide model relations that capture the NVD schema structure and the security exploit concepts extracted by the entity and concept spotter framework developed by Lal (9). We analyzed several cybersecurity-related blogs, security bulletins and CVE descriptions and identified a set of key classes defined in the ontology, specific to the entities which are part of the NVD dataset are *Vulnerability*, *Product*, *Attack Properties* and *Weakness*.

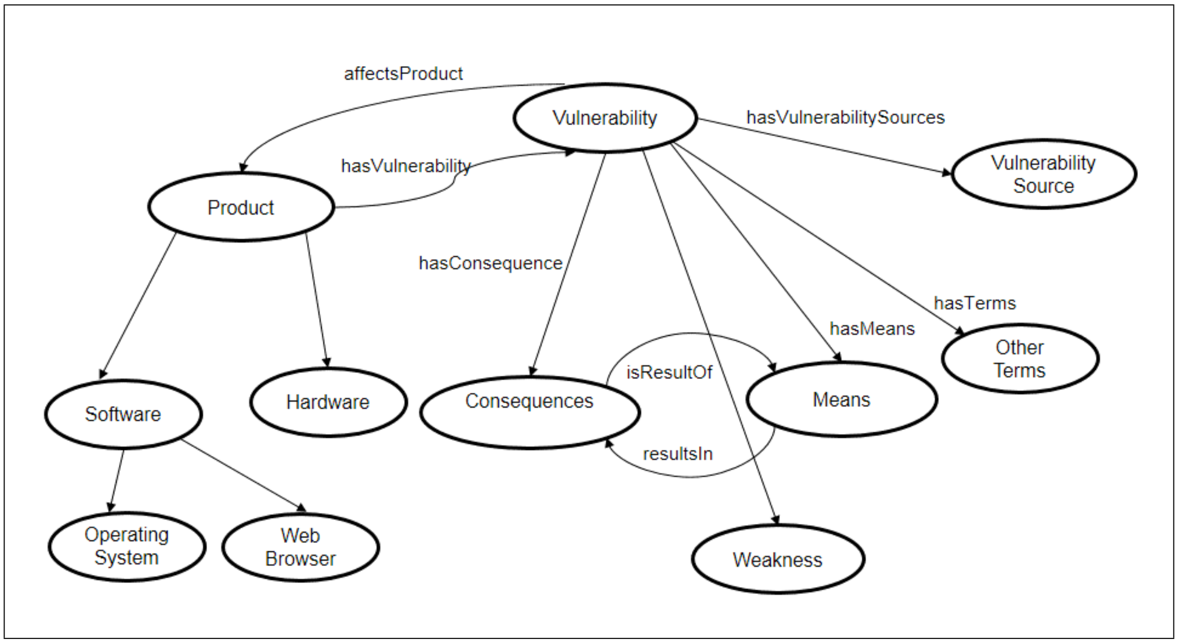


FIG. 3.2. A high level sketch of the IDS ontology

1. Vulnerability (The NVD/CVE entry)

    (a) Vulnerability Source (e.g: Secunia, Microsoft Security Bulletin)

2. Product

   (a) Hardware (e.g: Cisco Linksys router)

   (b) Software (e.g. Microsoft .NET Framework 3.5)

      i. Operating System (e.g. Ubuntu 10.4)

      ii. Web Browser (e.g: Google Chrome)

3. Attack Properties

   (a) Means: Way to attack (e.g. Buffer overflow)

   (b) Consequences: Final result of an attack (e.g. Denial of Service)

4. Weakness

3.3.1 Vulnerability

A vulnerability is an important class in the ontology, as each entry in the NVD is identified and documented based on a CVE number. The CVE number is a unique identifier for a vulnerability description provided by MITRE, on an incremental basis for each year. All information related to a particular identified vulnerability is reported based on the CVE ID such as the list of affected products (identified based on their unique CPE name[3]), the Web resources where it was first documented, and the severity metrics. The Vulnerability class hence is defined to have corresponding relationships with all other classes which are used to model entities that are part of an NVD entry.

Vulnerability Source The Vulnerability Source subclass captures the information provided in the NVD XML entry about the resource where the vulnerability was identified, or discussed. The sources of information are varied namely - technical reports, security blog entries such as Secunia or Bugzilla, and security bulletins from Adobe or Microsoft.

[3] Common Platform Enumeration - http://cpe.mitre.org/

These sources document the first hand information about the nature of the attack, including key sources such as the versions of an affected product, the known issues that result from such a vulnerability, the executive summary, and any remedial recommendations that are documented (based on initial analysis). Each source is tagged as one of the following category - “PATCH”, “THIRD_PARTY ADVISORY”, “VENDOR ADVISORY”, “MITIGATION PROCEDURE” or “UNKNOWN”. The NVD database asserts the vulnerability sources listed for each entry and these sources are confirmed to contain accurate information about the attack. Since most of these sources are maintained by vendors (e.g.: Microsoft) and established third party advisories (e.g.: Secunia), they are well-documented and are represented in a standardized structure.

3.3.2 Product

The Product class models the hardware and software products that are affected by a vulnerability. The Software subclass is further classified as an *Operating System* and *Web Browser* to correctly classify operating systems and web browsers, apart from a generic “application” tag. The affected product information described in an NVD entry is limited to a list of product names described for a particular vulnerability. This information is incorporated using the Common Platform Enumeration (CPE) format, which includes version granularity. The *affectsProduct* relationship models a one-many mapping between the vulnerability identifier and the list of affected products. Additional information about the affected products in the NVD entry is extracted using our cybersecurity entity and concept spotter.

3.3.3 Attack Properties

The idea behind modeling the *Attack Properties* class came from the work of Undercoffer et al. (11). An attack can be classified further as a *Means* or as a *Consequence*.

For example, “buffer overflow” is considered as an instance of a Means, since it is never an attacker’s final goal, but merely a step in achieving a significant consequence, such as “denial of service.”

Means This class mainly describes the immediate reaction of the system to the input. The means of vulnerability helps to identify a method of attack. The means consists of Input Validation Errors, Buffer Overows, Boundary Condition Errors and other Malformed Input.

Consequence This class describes the final result of an attack, and includes the categories of attack such as Denial of Service, Remote to Local and User to Root. The result of the attack is manifested as a Denial of Service, Unauthorized Access (user or root), Loss of Confidentiality and Information Leakage resulting from a probe.

3.3.4 Weakness

An NVD entry contains a unique CWE identifier which classifies a vulnerability, based on a hierarchy of attack classes modeled to generalize different attack signatures. For example, *Cross-side scripting (XSS)* is a subclass of *Code Injection* which is a subclass of the *Invalid Input*[4]. As described in Section 2.1.2, the severity score for a CVE ID is derived on parameters specified for the corresponding CWE ID. The *Weakness* class is thus included to extract more information regarding the metrics used to score the vulnerability’s severity, by which the means for addressing a threat will be refined and enhanced.

[4] CWE Hierarchy - http://nvd.nist.gov/cwe.cfm

3.4 Creating Ontological Properties

In the previous section, we identified concepts in the NVD dataset that capture semantics of the cybersecurity and product data. The final step in creating the IDS ontology was assigning domains and ranges to all of the object properties and data properties. This was complicated due to the deep nesting structure implemented in the NVD schema. Many attributes and sub-properties in a NVD entry had similar names and were “free-floating” that can be applied to multiple elements in the hierarchy. For example, the property “source” is used to denote a vulnerability source (security bulletin, technical blog) as well as for describing the source for the severity score (which is http://nvd.nist.gov for all current CVE descriptions, since all of them are scored using CVSS metrics). A similar case was found for the published date-time, modified date-time and generated date-time for a CVE description. All three properties have a date value, though are indistinguishable for a machine. All of these date related data properties were given ranges of `xsd:date`, which is a W3C standard format for expressing dates. Applying specific ranges, such as `xsd:date`, prevents catalogers from using inconsistent data formats and allows machines to parse and understand the data in the RDF output.

In order to add detail to the new NVD RDF data model, existing NVD XML elements needed to be parsed and converted into new ontological object properties. This was particularly a challenging aspect in designing the IDS ontology since the logical mapping of all relevant concepts from the NVD schema into the IDS ontology had to be maintained in order to not lose any context. Unfortunately, it is difficult for an automated framework to combine the individual pieces of an XML element to develop the same relationship between the XML element and the object being described for a vocabulary. For example, the list of affected products (in form of unique CPE identifiers for each product) was nested under `vuln:vulnerable-software-list` in the XML schema. We proposed an object property, `affectsProduct`, that maps all references to software products and includes CPE names as well as linked DBpedia resources for the corresponding software product.

Hash URIs

Protege allows for representing object properties and classes as hash URIs, with the vocabulary URI as the base. Hash URIs make use of a special part of a URI, the fragment, which is separated from the base part of the URI by a hash symbol "#". When a Web client looks up a hash URI, the HTTP protocol requires the fragment part to be stripped off before requesting this URI from a Web server. Thus, hash URIs can be used in Linked Data to identify real-world objects (non-information resources) without creating ambiguity. Hash URIs are used where it is needed to retrieve a single resource's description. Hence, whenever an object property or class defined in the IDS ontology is dereferenced, the entire vocabulary is retrieved.

3.5 RDF Representation of NVD

Semantics allow machine interpretation of links and relations between different properties of a vulnerability. Interlinking leads to an integrated and well-connected data corpus, available via an endpoint for advanced applications such as a semantic search and vulnerability statistics. Applying semantic web technologies to represent the data provided by the NVD dataset will be useful for semantic analysis of vulnerabilities and exploits. However, correlating this data to the existing concepts on the Web and reasoning over such a corpus is a vital task to avail this information for different applications, front-end services and data consumers (viz. security practitioners, system administrators).

As shown in Figure 3.1, our RDF-generation platform takes as input the NVD descriptions and associated properties from XML feeds provided for the NVD dataset, as described in Section 2.1.1. The framework then ingests this information and generates RDF triples via an Extensible Stylesheet Language Transformation (XSLT)[5] and the Jena RDF API[6]. For designing the XSL transformation, reference data provided from the NVD schemas (viz. `scap-core, cvss, cwe and vuln`) was initially used to better understand the structure of the NVD database and how they were formatted in XML. This was important because the XSLT stylesheet was designed to identify specific XML tags and convert them into RDF. Since the IDS ontology is custom built for a prototype, the relationships for mapping NVD XML attributes to the corresponding object properties were manually coded in the XSLT stylesheet.

The system extracts primary attributes included directly in the NVD schema, as well as advanced properties fetched from the sources described in the former. The CVSS schema includes attributes such as the *Access Vector*, *Access Complexity* and *Authentication*. These attributes are used to calculate the severity score for a threat. It is observed that the vulnerabilities with the same combination of these features, take place under the same context or running environment. In the RDF representation of the NVD data, these attributes are included as sub-properties to the `cvss base_metrics` property. By inclusion of these properties, we can enlist all the possible vulnerability descriptions that can affect a particular operating environment, by triggering a SPARQL query to return all the CVE descriptions with the same value for the `hasAccessVector`, `hasAccessComplexity` and `hasAuthentication` properties.

Furthermore, a NVD entry contains the CWE ID for the weakness class it belongs to. The CWE schema contains vital information which is mapped by our framework to the corresponding NVD entry, based on the CWE ID. In the attempt to retain as much

[5] http://www.w3.org/TR/xslt

[6] https://jena.apache.org/documentation/rdf/

valuable data as possible during the conversion process from XML to RDF, there was a need to analyze the schema, in order to determine what constituted as the valuable data. We include CWE-specific information in the NVD RDF graph such as

- description of the software weakness
- a hierarchical list of parent-child relationships for a weakness (e.g., *XSS* is a child of *Code Injection*, which is a child of the *Input Invalidation* category of weaknesses).
- Possible Consequences
- Mitigation Information
- Demonstrative Examples (if any)

Such information is essential to understand the background of an attack, and hence can be useful if represented in a condensed RDF format.

3.5.1 Representing Affected Product Information

As mentioned previously, the NVD dictionary provides a list of affected products as well as the vulnerable configurations that are possible for a vulnerability. The product information is represented using the CPE standard. Details such as the vendor information, the type of software product, the product name and the version are important metadata for the affected product and hence are represented as sub-properties for the *affectsProduct* object property. Based on the CPE name given for the affected product, we extract the vendor name, the software product, the type of software (e.g., Application, Operating Software) and the version number of the product. For example `cpe:/a:adobe:acrobat:_-reader9.5.3` denotes the application (`a`) provided by the vendor Adobe with the name Acrobat Reader and version 9.5.3. These pieces of information is represented as sub-properties of the *affectsProduct* object property. Deconstructing the CPE name into the above mentioned sub-properties helps identify software vulnerabilities based on a type of product, or offered by a particular vendor, and even distinguish between versions of a software product.

3.6 Publishing Linked Cybersecurity Data

The linked data publication step involves following the Linked Data principles. We based the selected publishing pattern on the existing NVD XML schema. This way, any updates made to the schema are easier to accommodate since they are just extensions to the ontology model. In addition to customizing the XSLT stylesheet to ensure that all of the important record data is converted into RDF, we attempted to introduce additional semantic depth to the data. This involved converting controlled vocabulary terms into deferenceable URIs. Establishing the relations between security exploit terms and resources on the Web that uniquely identify these concepts is essential to data integration. The objective to actually link instances and concepts with other data sources is a challenging aspect.

As shown in Figure 3.3, after the RDF instances are generated from the properties provided by the NVD schema, the link generation component of our framework connects the security concepts extracted from the vulnerability descriptions to the existing deferenceable resources on the Web. Each NVD entry mentions a short summary of the vulnerability description, which is essentially unstructured text. Our framework annotates security-related terms from the vulnerability description and maps them to corresponding DBpedia resources using DBpedia Spotlight[7]. DBpedia Spotlight is an annotation tool for finding mentions of DBpedia resources in free text. It links text mentions to Linked Open Data cloud instances through known (and deferenceable) DBpedia URIs. Moreover, DBpedia Spotlight provides flexibility to configure annotations to specific use cases, through

[7] spotlight.dbpedia.org

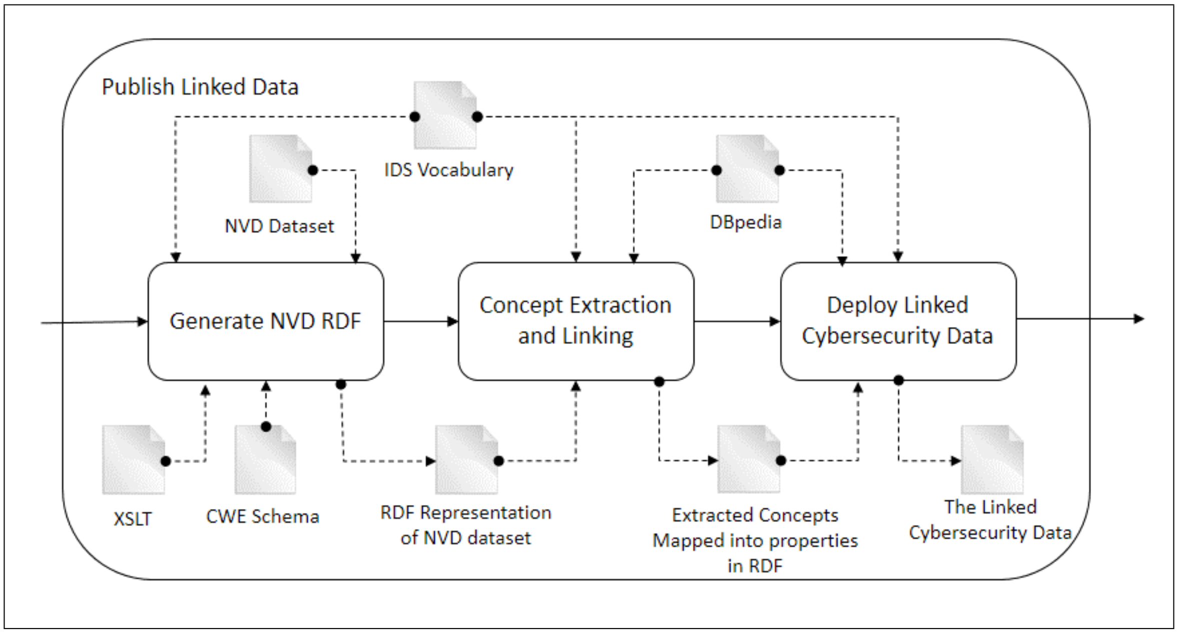

Publish Linked Data
IDS Vocabulary
NVD Dataset
DBpedia
Generate NVD RDF
Concept Extraction and Linking
Deploy Linked Cybersecurity Data
XSLT
CWE Schema
RDF Representation of NVD dataset
Extracted Concepts Mapped into properties in RDF
The Linked Cybersecurity Data

FIG. 3.3. Publishing NVD data as Linked Data process

quality metrics such as topical pertinence and disambiguation confidence(19). The best practices for publishing Linked Data recommend to reuse the terms of the existing vocabularies to define custom vocabularies (20). This will facilitate further reuse, uptake and interoperability of the dataset on the Web of Data. Binding the DBpedia references to the identified security concepts will enhance association of our linked data resource with the Linked Open Data cloud. Figure 3.4 gives an overview of the steps involved in publishing linked cybersecurity data, especially thr concept extraction and linking component, discussed in detail in the following sections.

3.6.1 Linking entities with DBpedia Spotlight

Entities with valid (contextual) resources in DBpedia are annotated based on adequate tuning of the confidence and support metrics. On experimentation of these parameters over our dataset, a confidence of 0.3 and a support of 20 generated relevant DBpedia links for

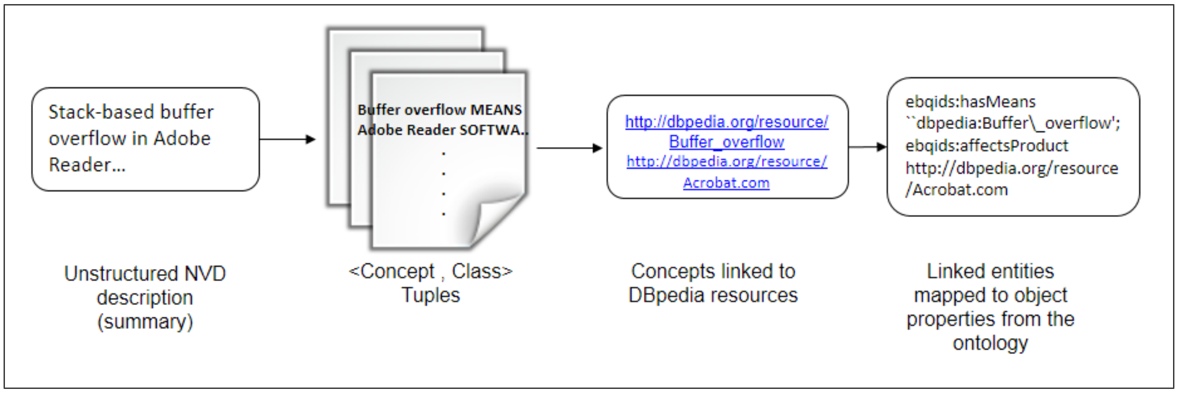

Stack-based buffer overflow in Adobe Reader...
Buffer overflow MEANS Adobe Reader SOFTWA..
http://dbpedia.org/resource/Buffer_overflow
http://dbpedia.org/resource/Acrobat.com
ebqids:hasMeans ``dbpedia:Buffer_overflow';
ebqids:affectsProduct http://dbpedia.org/resource /Acrobat.com
Unstructured NVD description (summary)
<Concept , Class> Tuples
Concepts linked to DBpedia resources
Linked entities mapped to object properties from the ontology

FIG. 3.4. The concept extraction and linking pipeline

Table 3.1. DBpedia Spotlight web service REST API call

Field Name	Definitions
http://spotlight.dbpedia.org/ rest/annotate	Runs spotting and disambiguation. Recognizes entities/concepts to annotate. Chooses an identifier for each recognized entity/concept given the context.
text	Content to be annotated
confidence	Provides the topical pertinence threshold
support	Returns the DBpedia entries which have *support* number of links or more

a specific vulnerability description. These values were based on the decision of selecting lesser, highly relevant entities against more, slightly relevant (tending towards irrelevant) terms being linked.

The annotations and subsequent linkages provided by DBpedia Spotlight are not final, or complete. It was observed that DBpedia Spotlight tends to miss out on a considerable amount of security-related terms, especially the types of attacks and special terms. The problem can be attributed to the terms described in free text in a manner which is different to the corresponding DBpedia resource. For example, the CVE description text had mentions of "*Arbitrary code execution*", that has a related DBpedia resource, though was represented as "*...executed arbitrary code...*". This inadequacy can be attributed to the higher values given to the confidence and support metrics, as justified previously.

In addition to missing relevant terms from text, DBpedia Spotlight also posed challenges while mapping the terms to concepts and relevant object properties from the IDS ontology. For a given piece of text, the DBpedia Spotlight API returns the sets of annotated terms and corresponding DBpedia resources (URIs). However, the annotation does not provide the corresponding class from the DBpedia ontology that the resource belongs to.

Both of the above mentioned problems were tackled using the cybersecurity entity and concept spotter, developed by Lal et al., to link spotted security exploit concepts to relevant sources on DBpedia via DBpedia Spotlight, and to map these links to appropriate classes from the IDS vocabulary. The cybersecurity entity and concept spotter uses a Conditional Random Field (CRF)-based classifier that helps filter relevant concepts and entities from the text (9). The extraction framework was trained over several cybersecurity-related blogs, security bulletins and CVE descriptions. The NVD descriptions (`vuln summary`) are passed through the concept spotter, that identifies relevant terms, assigns a class label, and returns a set of `<Concept, Class>` tuples for the description.

All the *Concept* terms from `Concept, Class` annotated pairs for a NVD description, returned by the concept spotter, are then passed on through the DBpedia Spotlight API. DBpedia Spotlight provides a DBpedia link (http://dbpedia.org/resource) for each term it finds on the DBpedia knowledge base in the form of a “DBpedia Annotations” XML response with the DBpedia URI as part of a `<a href>` tag. The spotted terms from the description that do not return a DBpedia link are ignored, based on the assumption that the term does not have a relevant DBpedia resource. The justification for this approach is provided in Chapter 4. This approach is based on the performance of the entity and concept spotter, combined with the DBpedia Spotlight web service; compared to the standalone usage of DBpedia Spotlight.

3.6.2 Mapping controlled vocabulary properties to DBpedia concepts

The annotated terms from DBpedia Spotlight were compared against the entities identified by the concept spotter using a string comparison. The corresponding DBpedia resource for the matched concept is assigned a class value, based on the *Concept, Class* pairs. Further, based on the assigned class label for the concept URI, these resources are then mapped with an appropriate object property from the IDS vocabulary.

For example, Figure 3.5 demonstrates working of the concept extraction and linking framework for a single NVD entry. The concept spotter identifies relevant terms from the NVD description, such as "Buffer Overflow" and provides an appropriate class label "Means". The pair `<Buffer Overflow, Means>` is passed through the DBpedia Spotlight API to generate a DBpedia resource URI for Buffer Overflow, if available. The framework then assigns the corresponding object property it possesses with the Vulnerability class as the subject and/or the object. In this case, the framework assigns the object property *hasMeans* to "http://www.dbpedia.org/resource/Buffer_overflow" for the corresponding NVD entry ID. Similarly, the list of affected products are listed with the *affectsProduct* property. Those annotations which do not fall under any class that is modeled as part of the IDS ontology, and has an associated object property with the Vulnerability class, are included in the RDF graph as part of the *hasTerms* relationship. This is important since many basic concepts such as *HTTP*, *dll* do not fall under a concrete category, though can be useful while searching a vulnerability in the RDF graph. The SPARQL query with the filter of the *hasTerms* property can be used to narrow the search for a particular (or a group) of vulnerability description(s).

The IDS vocabulary models key aspects of a cyber attack which are not represented precisely in the DBpedia ontology. For example, the terms *Buffer Overflow* and *Denial of Service* are aptly represented as "Means" and "Consequence" respectively in the IDS

CVE-2012-0150
Buffer overflow in msvcrt.dll in Microsoft Windows Vista SP2, Windows Server 2008 SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 allows remote attackers to execute arbitrary code via a crafted media file, aka ”Msvcrt.dll Buffer Overflow Vulnerability.”

```
Buffer Overflow MEANS
dll FILE
Microsoft Windows Vista OPERATINGSYSTEM
SP2 NER_MODIFIER
Windows Server OPERATINGSYSTEM
SP2 NER_MODIFIER
R2 NER_MODIFIER
R2 NER_MODIFIER
SP1 NER_MODIFIER
Windows 7 OPERATINGSYSTEM
execute arbitrary code CONSEQUENCE
file FILE
dll FILE
Buffer Overflow MEANS
Vulnerability OTHER
```

```
http://dbpedia.org/resource/Buffer_overflow, http://dbpedia.org/resource/Dynamic-link_library,
http://dbpedia.org/resource/Windows_Vista,http://dbpedia.org/resource/Windows_7,
http://dbpedia.org/resource/Arbitrary_code_execution,http://dbpedia.org/resource/Computer_file
,http://dbpedia.org/resource/Dynamic-link_library,http://dbpedia.org/resource/Buffer_overflow,
http://dbpedia.org/resource/Vulnerability_(computing)
```

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ebqids: <http://ebiquity.umbc.edu/IDSv2.0.1.owl#> .
<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-0150>
ebqids:affectsProduct
“http://dbpedia.org/resource/Windows_Vista” , “http://dbpedia.org/resource/Windows_7” ;
ebqids:hasMeans
“http://dbpedia.org/resource/Buffer_overflow” ;
ebqids:hasConsequence
“http://dbpedia.org/resource/Arbitrary_code_execution” ;
ebqids:hasTerms
“http://dbpedia.org/resource/Computer_file” , “http://dbpedia.org/resource/Dynamic-link_library”
, “http://dbpedia.org/resource/Vulnerability_(computing)” .
```

FIG. 3.5. Concept extraction and linking framework (a) The description section of NVD entry CVE-2012-0150. (b) Extracted concepts and corresponding class labels, identified by the concept spotter. (c) Relevant DBpedia URIs for the description, returned by passing the annotated concepts through DBpedia Spotlight. (d) Turtle output. Based on the class label assigned to the tagged entities, the DBpedia resources are mapped to relevant object properties from the IDS ontology.

vocabulary. These concepts are highly specific to a domain and hence not modeled in the DBpedia ontology.

Figure 3.6 shows a sample NVD entry which specifies the CVE identifier for the vulnerability description, together with the list of affected products with CPE names, the Weakness identifier, and the source where the vulnerability was documented. We extract information from this data to generate machine- understandable assertions in RDF, as shown in Figure 3.7. We use the IDS ontology to interpret key security concepts such as the vulnerability sources and severity metrics. Besides modeling semi-structured information, our framework extracts relevant DBpedia resources from the text description such as *Arbitrary_code_execution* and maps them to appropriate relationships from the IDS vocabulary. Based on the relationships established with the linked concepts, we can retrieve vulnerabilities and attack descriptions pertaining to a specific product version, those affected by a specific means (*Buffer Overflow*), or those attacks that are carried out under the same operating environment.

3.6.3 Linked Data validation and manual review

As a final check to make sure that the XSLT stylesheet was working properly and generating the correct RDF, the RDF/XML output was run through the W3C RDF validation service[8]. The NVD RDF graph was also validated using Linked Data validator tools such as Hyperthing[9], Vapor (21) and URI Debugger[10].

Although these services are not able to evaluate the quality of the RDF, it nevertheless was an important first step in determining how successful the XSLT stylesheet was at converting XML in RDF. The completed stylesheet was complete, was run against the entire

[8] http://www.w3.org/RDF/Validator/

[9] http://www.hyperthing.org/

[10] http://linkeddata.informatik.hu-berlin.de/uridbg/

4696 record NVD/CVE data set for the year 2013. The output then was manually reviewed in order to ensure that the XSLT templates worked and produced meaningful RDF. Figures 3.8, 3.9, 3.10 show the screen shots of validating a sample NVD RDF graph of 5 NVD entries via Hyperthing, Vapour and URI debugger respectively.

3.6.4 Pushing linked cybersecurity data through RSS feeds

The NVD datasets provide an RSS data feed (22) on all recent CVE vulnerabilities. The RSS feeds and "recent" and "modified" XML files (nvdcve-recent.xml and nvdcve-modified.xml) are automatically updated every 2 hours. The "recent" feed includes all recently published vulnerabilities, while "modified" RSS feed includes all recently published and recently updated vulnerabilities. These immediate data sources provide an ad-hoc report of the vulnerability, and can prove beneficial in reducing the time for threat detection and resolution. Both of these feeds can be represented as machine-understandable assertions as shown above.

Such RDF assertions can be added to the triple store, and can help in applications such as a situation aware intrusion detection system that can consume linked data to generate rules and alerts on possible threats. In the future, we plan to extend the concept spotting system into an information extraction framework that is not limited to the NVD dataset and its auxiliaries. The proposed system will extract concepts from free text, find relationships between entities spotted in the text, make assertions about them based on a specific heuristic and publish it to the linked cybersecurity data resource.

3.7 Deploying Linked Cybersecurity Data

The basic means to access Linked Data on the Web is to dereference HTTP URIs in RDF descriptions. There are two alternative ways to make Linked Data sets accessible on the Web: RDF dumps and SPARQL endpoints. The RDF dump of a Linked Data set is simply an RDF file (or multiple RDF files) that contains the RDF graph which describes the entire dataset. The RDF dump of a Linked Data set is simply an RDF file (or multiple RDF files) containing the RDF graph which describes the whole dataset. For the linked cybersecurity data, the set of NVD RDF files are hosted on the Ebiquity website[11], available to download. In this section we describe querying the NVD RDF graph using SPARQL, the language to query RDF.

We provide a SPARQL endpoint for querying the physical NVD RDF storage. We used Apache Jena Fuseki[12] that provides a HTTP-based query service that can be accessed using the SPARQL protocol. Fuseki follows the W3C standard method for remote invocation of SPARQL queries over the Linked Cybersecurity Data set. Fuseki has a built-in version of Jena TDB[13] that is used for high-performance RDF storage, and can be invoked via the the Fuseki server using the Apache Jena API. Fuseki returns data in the standard SPARQL Query Results XML format, and an XSLT stylesheet included with Fuseki formats it for display in the web browser.

The usability and discoverability of a linked data resource is directly dependent on the format in which the data that is modeled in the collection, and is made available to the appropriate consumers. In case of the linked cybersecurity data resource, the consumers are security practitioners, system administrators, or even the general population trying to be aware about the possible threats that might affect her system. This aspect of discoverability can be overcome via a SPARQL front-end. We plan to use Pubby (23) - a Linked Data interface to SPARQL endpoints. By providing an interface for Linked Data clients such as HTML and RDF browsers, we are presenting the data that is suitable for human

[11] http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/

[12] https://jena.apache.org/documentation/serving_data/

[13] http://jena.apache.org/documentation/tdb/

interpretation, in contrast to providing raw RDF triples as the output. We also incorporated YASGUI[14], a SPARQL user interface for our Fuseki SPARQL endpoint.

Figure 3.11 shows a sample SPARQL query passed to the NVD RDF graph that returns all NVD entries that have *Buffer overflow* as means.

Similar, more complex queries can be triggered to return more filtered results, providing a utility to search the NVD RDF graph more effectively. Linked Data enables distributed SPARQL queries of the data sets and a browsing or discovery approach to finding information (as compared to a search strategy). We can query over such a knowledge base via SPARQL queries to avail statistics on vulnerability trends, and can view the past history associated with a vulnerability or a particular software product. A triple store of such condensed information facilitates for a rich linked data resource, that can be used for semantic analysis of vulnerabilities.

There is a need to separate URIs to identify real-world entities and documents describing them, so that they are not mixed up. Linked Data provides two approaches : hash URIs and 303 URIs. Both approaches allow to make dereferenceable URIs of the real world entities and distinguishable from the URIs of the documents themselves. In the future, we plan to use the URL-rewriter mechanism provided by Virtuoso that allows dynamic generation of the RDF descriptions of the requested HTTP URIs describing security concepts from the IDS vocabulary and the NVD dataset.

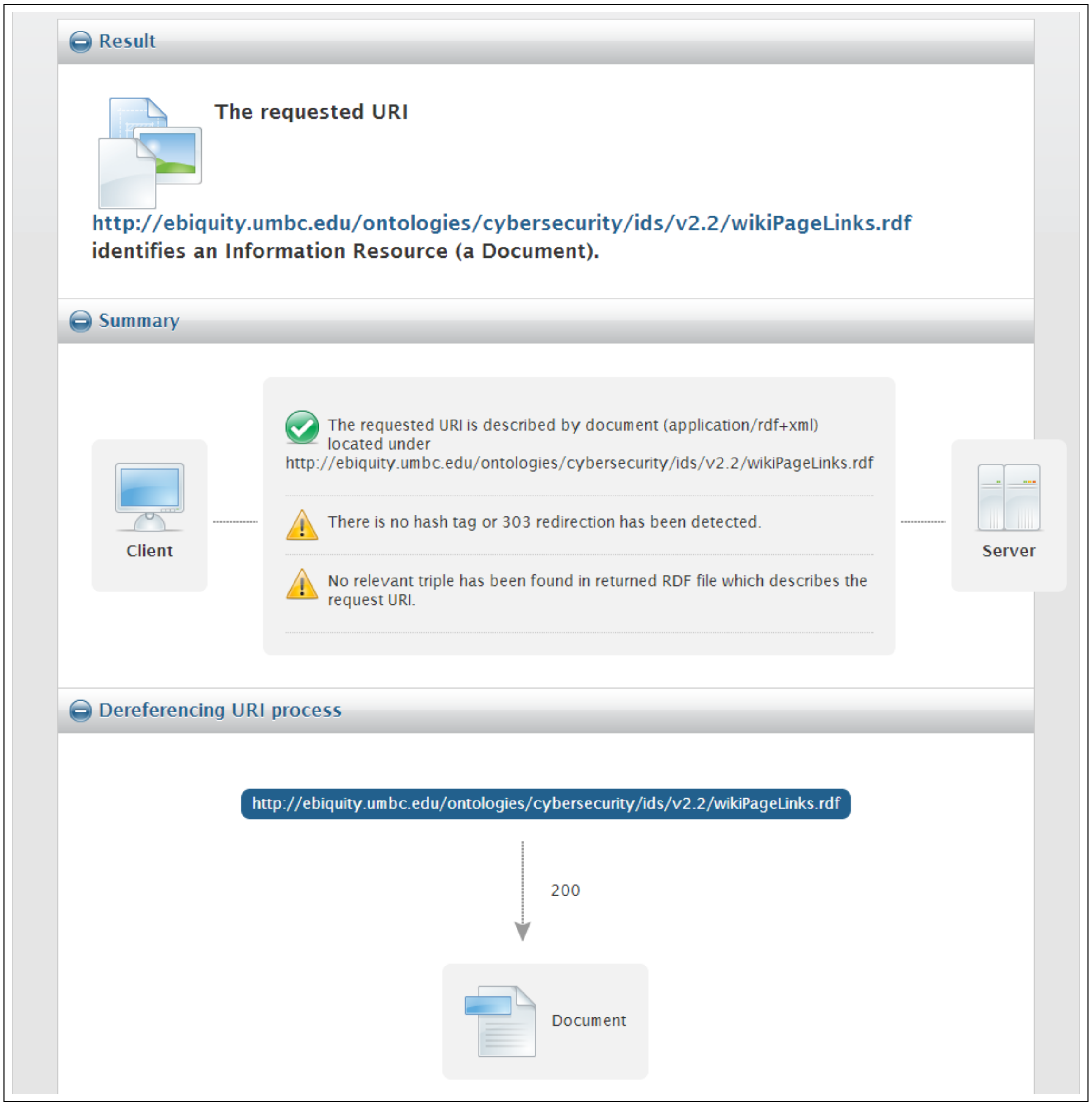
[14] http://yasgui.laurensrietveld.nl/

```
<?xml version="1.0" encoding="UTF-8"?>
<nvd xmlns:vuln="http://scap.nist.gov/schema/vulnerability/0.4"
xmlns:cvss="http://scap.nist.gov/schema/cvss-v2/0.2">
<entry id="CVE-2012-0150">
<vuln:vulnerable-software-list>
<vuln:product>cpe:/o:microsoft:windows_vista::sp2:x64 </vuln:product>
<vuln:product>cpe:/o:microsoft:windows_7:::x86 </vuln:product>
<vuln:product>cpe:/o:microsoft:windows_7::sp1:x86 </vuln:product>
<vuln:product>cpe:/o:microsoft:windows_vista::sp2 </vuln:product>
</vuln:vulnerable-software-list>
<vuln:cve-id>CVE-2012-0150</vuln:cve-id>
<vuln:cvss>
<cvss:base_metrics>
<cvss:score>9.3</cvss:score>
<cvss:access-vector>NETWORK</cvss:access-vector>
<cvss:access-complexity>MEDIUM</cvss:access-complexity>
<cvss:authentication>NONE</cvss:authentication>
</cvss:base_metrics>
</vuln:cvss>
<vuln:cwe id="CWE-119" />
<vuln:references xml:lang="en" reference_type="VENDOR_ADVISORY">
<vuln:source>MS</vuln:source>
<vuln:reference href="http://technet.microsoft.com/security/bulletin/MS12-013"
xml:lang="en">MS12-013</vuln:reference>
</vuln:references>
<vuln:summary>Buffer overflow in msvcrt.dll in Microsoft Windows Vista SP2,
Windows Server 2008 SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 allows
remote attackers to execute arbitrary code via a crafted media file,
aka "Msvcrt.dll Buffer Overflow Vulnerability."
</vuln:summary>
</entry>
</nvd>
```

FIG. 3.6. An excerpt of an NVD XML entry

```
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ebqids:<http://ebiquity.umbc.edu/IDSv2.0.1.owl#> .
@prefix dbpedia:<http://dbpedia.org/resource/> .
<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-0150>
ebqids:cveID “http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0150” ;
ebqids:cweID “http://cwe.mitre.org/data/definitions/119” ;
ebqids:affectsProduct “dbpedia:Windows_Vista” , ”dbpedia:Windows_7” ;
ebqids:summary “Buffer overflow in msvcrt.dll in Microsoft Windows Vista SP2, Windows
Server 2008 SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 allows remote attackers
to execute arbitrary code via a crafted media file,
aka “Msvcrt.dll Buffer Overflow Vulnerability.”” ;
ebqids:hasAccessComplexity “MEDIUM” ;
ebqids:hasAccessVector “NETWORK” ;
ebqids:hasAuthentication “NONE” ;
ebqids:hasSeverityScore “9.3” ;
ebqids:hasVulnerabilitySource
“http://technet.microsoft.com/security/bulletin/MS12-013” ;
ebqids:hasMeans “dbpedia:Buffer_overflow” ;
ebqids:hasConsequence “dbpedia:Arbitrary_code_execution” ;
ebqids:hasTerms “http://dbpedia.org/resource/Computer_file” ,
“http://dbpedia.org/resource/Dynamic-link_library” ,
“http://dbpedia.org/resource/Vulnerability_(computing)” .
```

FIG. 3.7. Turtle representation of extracted information



The requested URI

http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf identifies an Information Resource (a Document).

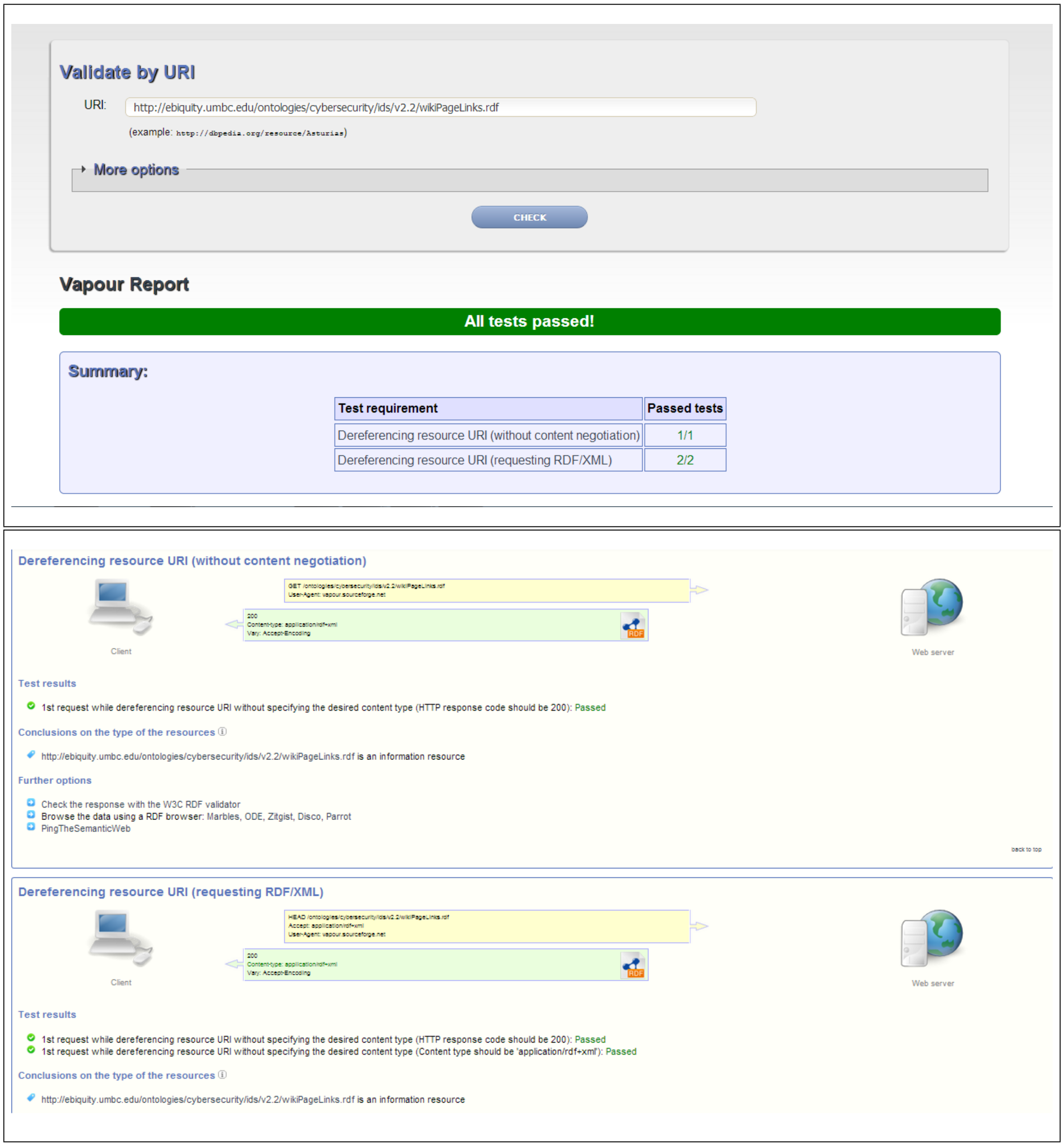
✓ The requested URI is described by document (application/rdf+xml) located under http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf

⚠ There is no hash tag or 303 redirection has been detected.

⚠ No relevant triple has been found in returned RDF file which describes the request URI.

http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf

FIG. 3.8. Validation of sample NVD RDF graph via Hyperthing Linked data validator



Validate by URI

URI: http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf

(example: http://dbpedia.org/resource/Asturias)

Vapour Report

All tests passed!

Summary:

Test requirement	Passed tests
Dereferencing resource URI (without content negotiation)	1/1
Dereferencing resource URI (requesting RDF/XML)	2/2

Dereferencing resource URI (without content negotiation)

```
GET /ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf
User-Agent: vapour.sourceforge.net
```

```
200
Content-type: application/rdf+xml
Vary: Accept-Encoding
```

Test results

- 1st request while dereferencing resource URI without specifying the desired content type (HTTP response code should be 200): Passed

Conclusions on the type of the resources ⓘ

- http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf is an information resource

Further options

- Check the response with the W3C RDF validator
- Browse the data using a RDF browser: Marbles, ODE, Zitgist, Disco, Parrot
- PingTheSemanticWeb

Dereferencing resource URI (requesting RDF/XML)

```
HEAD /ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf
Accept: application/rdf+xml
User-Agent: vapour.sourceforge.net
```

```
200
Content-type: application/rdf+xml
Vary: Accept-Encoding
```

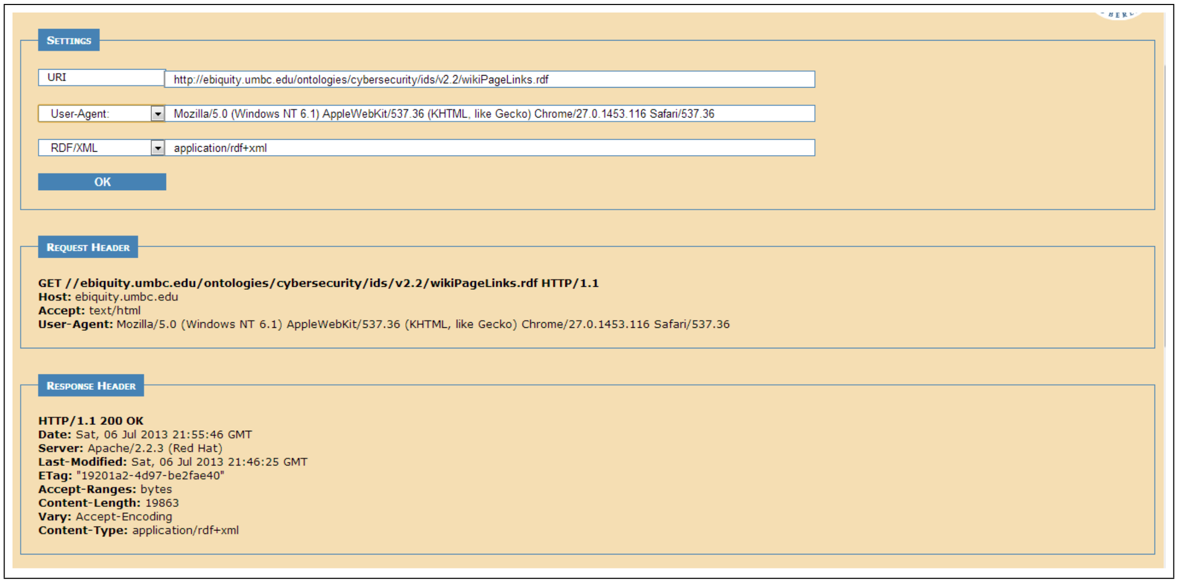
Test results

- 1st request while dereferencing resource URI without specifying the desired content type (HTTP response code should be 200): Passed
- 1st request while dereferencing resource URI without specifying the desired content type (Content type should be 'application/rdf+xml'): Passed

Conclusions on the type of the resources ⓘ

- http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf is an information resource

FIG. 3.9. Validation of sample NVD RDF graph via Vapour Linked data validator



```
URI          http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf
User-Agent:  Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36
RDF/XML      application/rdf+xml
```

```
GET //ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/wikiPageLinks.rdf HTTP/1.1
Host: ebiquity.umbc.edu
Accept: text/html
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36
```

```
HTTP/1.1 200 OK
Date: Sat, 06 Jul 2013 21:55:46 GMT
Server: Apache/2.2.3 (Red Hat)
Last-Modified: Sat, 06 Jul 2013 21:46:25 GMT
ETag: "19201a2-4d97-be2fae40"
Accept-Ranges: bytes
Content-Length: 19863
Vary: Accept-Encoding
Content-Type: application/rdf+xml
```

FIG. 3.10. Validation of sample NVD RDF graph via URI Debugger

```
# filename: nvdcve-2013.ttl
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ebq: <http://ebiquity.umbc.edu/ids/IDSOntology.owl#>
SELECT ?vuln
WHERE { ?vuln ebq:hasMeans “http://dbpedia.org/resource/Buffer_overflow” .}
```

FIG. 3.11. A sample query for the NVD RDF graph to return all entries that have “Buffer overflow” as the means

Chapter 4

SYSTEM EVALUATION

The focus of this study has been on generating a quality linked data resource for concepts described in the NVD dataset, and extracting and mapping cybersecurity terms, entities and relations from associated dictionaries. There have been continuing efforts and discussions among the Linked Data community regarding the possible methods of evaluation for a linked data collection. Flemming (24) proposed a set of criteria to assess the quality of Linked Data sources. The study specifies four categories: Content, Representation, Usage and System, that are used as reference to qualitatively analyze a Linked Data set. However, these criteria focus mainly on the publication of instance data and do not describe much about the schema level. The quality metrics are better suited for knowledge bases that cover several genres of data. In our case, the vocabulary is limited to the cybersecurity domain and hence these metrics do not qualify. Further, the quality assessments represent a check list for linked data generation, and are elicited on the lines of the best practices for Linked Data (5-star principles).

In this section an evaluation method is presented including a discussion of results. The evaluation is focused on assessing the quality measurements of linked data, especially the extracted concepts from the unstructured pieces of text of the NVD dataset. The retrieval results are compared to a ground truth resulting in an objective assessment of the achieved quality, i.e. via statistical analysis such as precision and recall. Since the efficacy of Linked Data usage and the quality of representation depends strongly on a qualitative measurement, such as user experience and satisfaction, it cannot be metricized. However, the efficacy of content in the linked data collection, and the accuracy of the system can be evaluated.

4.1 Concept Extraction and Linking

We calculated the precision and recall measurements for the concept extraction and linking framework. As described in the previous section, the concept extraction framework by Lal (9) identifies security concepts and terms and assigns an appropriate class label for the same. This concept spotter was trained over a data corpus of unstructured texts from security blogs, CVE descriptions and security bulletins.

Although, the concept spotter is capable of recognizing and annotating a higher percentage of cybersecurity-related terms from the NVD/CVE description text, it does not link them to dereferenceable URIs describing the concept. There is a considerable difference in the number of annotations picked by the cybersecurity concept spotter and the number of annotations (and thereby links) generated by DBpedia spotlight. It has been observed that DBpedia Spotlight occasionally does not spot concepts, although they are included in the DBpedia dictionary (concept has a DBpedia resource). Figure 4.1 shows the comparison for the number of annotations extracted from a set of 300 NVD vulnerability descriptions by DBpedia spotlight and the concept extraction framework. This inadequacy is overcome by using DBpedia Spotlight on top of this extraction framework.

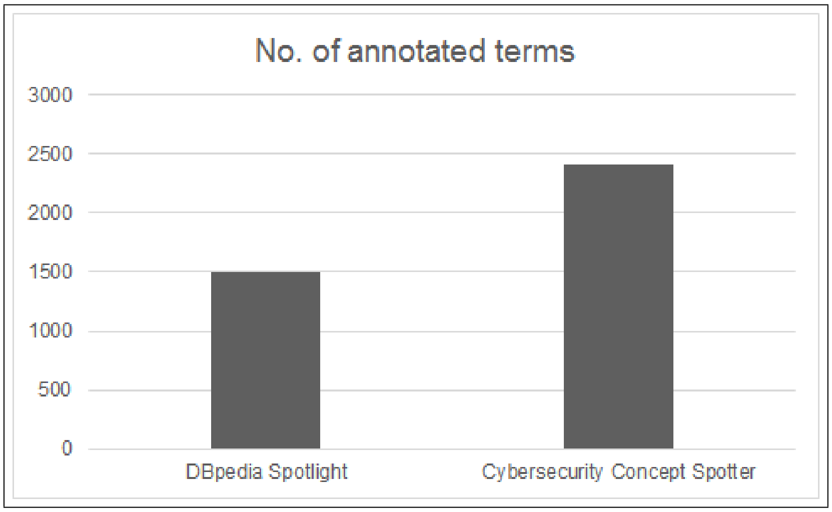


FIG. 4.1. Comparison of number of annotations

Table 4.1. Aggregate result comparisons considering the average of the precision and recall

	AlchemyAPI	DBpedia Spotlight	Extractiv	OpenCalais	Zemanta
overall precision	0.7054	0.4915	0.611	0.5396	0.6463
relevant score	0.9005	0.5525	0.6805	0.8224	0.8800

4.1.1 Choosing DBpedia Spotlight

We decided on using DBpedia Spotlight for the link generation component of our framework, based on the results derived from the study by Rizzo et al. (25). Several named entity (NE) extractors such as DBpedia Spotlight, Alchemy API[1], Extractiv[2], OpenCalais[3] and Zemanta were compared for their overall performance, as also for URI disambiguation. Table 4.1 (25) gives the result of the comparison for all NE extractors.

It was observed that DBpedia Spotlight presents substantial agreement for URI disambiguation. Alchemy API, although preserving good performance in NE extraction and

[1] http://www.alchemyapi.com/

[2] http://extractiv.com/

[3] http://www.opencalais.com/

accurate typing, has a clear weakness to link the NE to a web resource. DBpedia Spotlight was observed to perform better in all of these paradigms. It should be noted that the usage of DBpedia Spotlight in our framework is purely for link generation. The entity recognition component is handled by our concept extractor. Further, most of the NE extractors are trained to identify *People*, *Places* and *Organization*, and fail to identify concepts that are specific to the cybersecurity domain. Based on this need, DBpedia fares better than most of the available NE frameworks. Moreover, DBpedia Spotlight has a large, dynamically updated database (Wikipedia) and these updates are pushed simultaneously to the DBpedia knowledge base. Hence, DBpedia Spotlight was the clear choice as a link generation component, from among the above mentioned NE extraction frameworks.

4.1.2 Evaluation of Concept Extraction and Linking

In order to asses the quality of the extracted concepts and linked entities (DBpedia resources) that are mapped to these annotations, we evaluated the system against the ground truth. The ground truth was generated via human annotations for a set of 300 NVD/CVE descriptions. The 300 NVD/CVE descriptions were picked up from the most updated, June 2013 XML from the NVD repository. The descriptions consisted of about 1836 spotted concepts that were linked to DBpedia resource URIs. These descriptions were passed through the cybersecurity entity and concept spotter, and the spotted concepts were passed through DBpedia Spotlight to generate DBpedia resource URIs for the same. The plain text content for each NVD/CVE description, annotated with the linked entities, was provided to human annotators. The annotators were Graduate students from the Computer Science department at UMBC. The annotators were assumed to have a fair knowledge and understanding about the terms and concepts that are relevant to the cybersecurity domain.

Calculating Precision In the world of information retrieval, precision is "the fraction of retrieved instances that are relevant" (26). Precision can be calculated as follows.

(4.1)

$$Precision(per\,NVD/CVE\,description) = \frac{\text{Number of relevant links (TP)}}{\text{Total no. of annotated concepts (TP + FP)}}$$

In our case, the relevant terms refer to those security-related concepts that are picked up by the concept spotter and linked to the appropriate DBpedia resource. For calculating the precision for the system, the annotators provided a score of 1 (highly relevant), 0.5 (moderately relevant, enough to represent the entire term in text) or zero(0) (irrelevant, wrong annotation) for every NVD/CVE description text in the test set. The total score of these scores was considered the set of *True Positives*(TP), and divided over by the total number of annotations in a given piece of text (*True Positives* + *False Positives*(FP)).

Calculating Recall Recall is defined as "the fraction of relevant instances that are retrieved" (26). Recall is calculated as follows.

(4.2)

$$Recall(per\,NVD/CVE\,description) = \frac{\text{Number of relevant links (TP)}}{\text{Total no. of annotated concepts expected(TP + FN)}}$$

The total number of expected annotations was calculated with the help of annotators. The annotators were asked to identify those terms in the text that should have been annotated to a relevant DBpedia resource but were missed by the concept spotter, and subsequently by DBpedia Spotlight in separate degrees of variation. In order to aid the annotators in identifying such terms, we provided the `<Concept, Class>` tuples that were generated by the concept spotter, but missed by DBpedia Spotlight (these terms were ignored with the assumption that they do not have a relevant DBpedia resource). The annotators provided a score of 1 for those terms that they felt are expected and are correctly annotated by our mash-up framework. A score of zero (0) indicated, that the expected link was missing or incorrect. Identifying these concepts was subject to the restriction that the terms to be annotated had a matching Wikipedia page (which ensures the presence of a corresponding DBpedia resource). This indicates that DBpedia Spotlight did not identify the term, and hence constitutes as a zero.

For those terms that were given a score of 0.5 in the precision calculation, the annotators were given the discretion to choose the linking as relevant, only if they felt that the annotated part of the concept was a representative of the entire concept. For example, consider a text having a description "Microsoft Windows XP SP2". If the concept spotter identifies the term *Windows*, the annotator would choose a 1 for the concept (i.e. a strong match). However, had the concept spotter identified *Microsoft*, the annotator would choose a 0, since the term "Microsoft" does not entirely represent the term described in text.

Further, the annotators were asked to identify those annotated terms that should have not been annotated. This indicates that the annotators felt a particular link is incorrect and should have not been identified as a cybersecurity-related term. Based on these parameters, a total count of the expected annotations and links for a given NVD/CVE description document.

Based on the scores given on the annotations, and count of expected terms that should have been extracted and subsequently linked to the correct DBpedia resource, the precision and recall for the concept extraction and linking component were calculated. Figure 4.2 shows the comparison of the performance of using the standalone DBpedia Spotlight web service on the NVD/CVE description text, versus passing the text through the pipeline

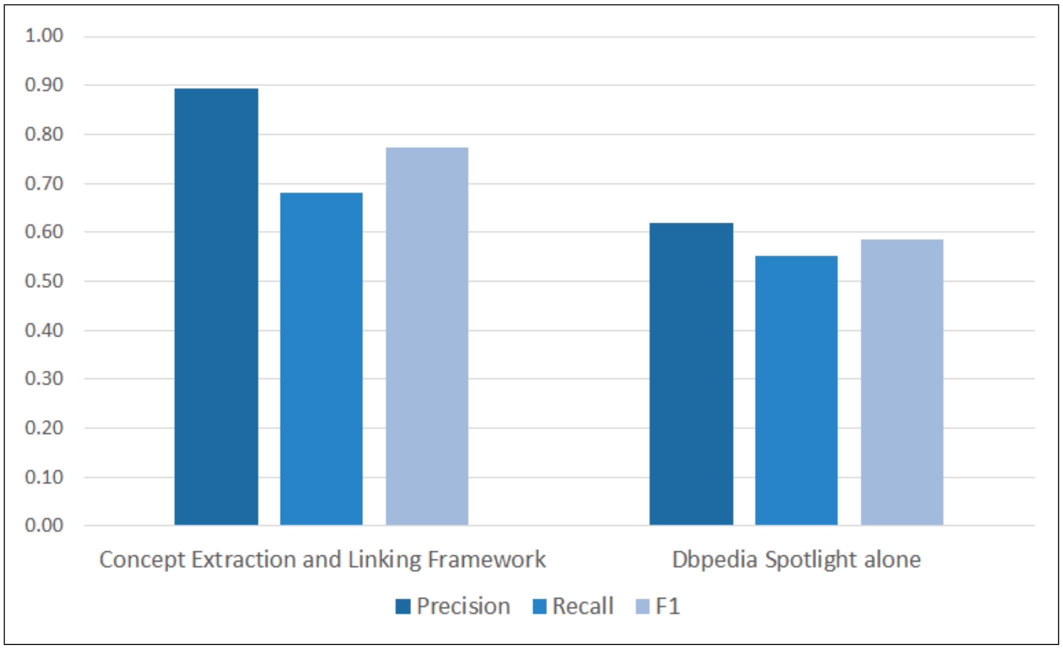


FIG. 4.2. Overall performance of the concept extraction and linking framework

of extracting concepts with the concept spotter and then through DBpedia Spotlight, with suitable confidence and support metrics.

The average precision value was found to be 0.893, the average recall to be 0.638 and the F1 score was 0.74. It was observed that the precision is higher than using DBpedia Spotlight alone. As described in the previous section, this can be attributed to the fact that DBpedia Spotlight was observed to be unable to identify a concept when it is represented in plain text, in a manner that is different to the appropriate term in the DBpedia knowledge base. Moreover, the training dataset for DBpedia Spotlight was very diverse than is the case for the CRF-based classifier utilized by the concept spotter.

The recall for both systems was observed to be comparable, indicating that most of the entities spotted by the concept spotter but not linked DBpedia Spotlight, may not have a relevant entry in the DBpedia knowledge base. In the future, the recall can be improved for this system, by enabling a dynamic dereferenceable resource generation framework, that can link the extracted concept not identified by DBpedia Spotlight, to a document on the Web that describes the resource. The assumption of ignoring those terms extracted by the concept spotter, though not linked to a relevant DBpedia resource by DBpedia Spotlight, is based on the performance demonstrated by combining the spotter component with DBpedia Spotlight. The combination returns a much higher precision and overall performance than the standalone DBpedia web service. Hence, assuming that the missed entries do not have a relevant DBpedia resource is justified.

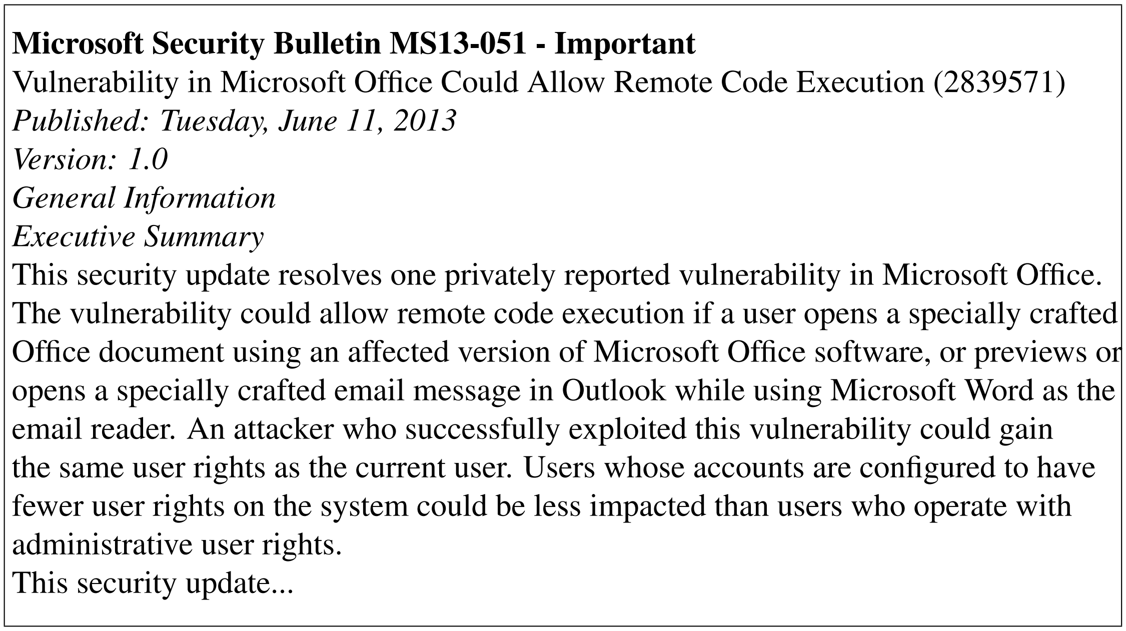
4.1.3 Concept Extraction and Linking on Non-NVD/CVE data

The scope of the thesis was limited to generating linked data for the concepts and entities that are described in the NVD dataset. However, in order to analyze the effectiveness of our system on data which is not part of the NVD dataset, we also tested the system on documents from other sources of vulnerability reporting and management. The experiments open the possibilities of inclusion of documents on the Web that are not as structured as the NVD dataset, though can be linked to concepts modeled in the Linked Cybersecurity Data.

We collected 50 documents from various sources of ad-hoc vulnerability reporting - 30 SecurityFocus [4] documents, 10 from TechNet (Microsoft Security Bulletins portal) and 10 from Adobe Security Bulletins. SecurityFocus is a mailing list for reporting vulnerabilities, an online computer security news portal and purveyor of information security services, and home to the well-known Bugtraq[5] mailing list. Microsoft and Adobe Security Bulletins, provide a detailed executive summary of a vulnerability, the affected software, recommendation for resolution and other details. A common factor in these documents is the that the style of documentation, as well the description of the relevant concepts is different, compared to the NVD/CVE description. Figures 4.3, 4.4 and 4.5 show excerpts

[4] http://www.securityfocus.com/

[5] http://www.securityfocus.com/archive/1/description#0.1.1

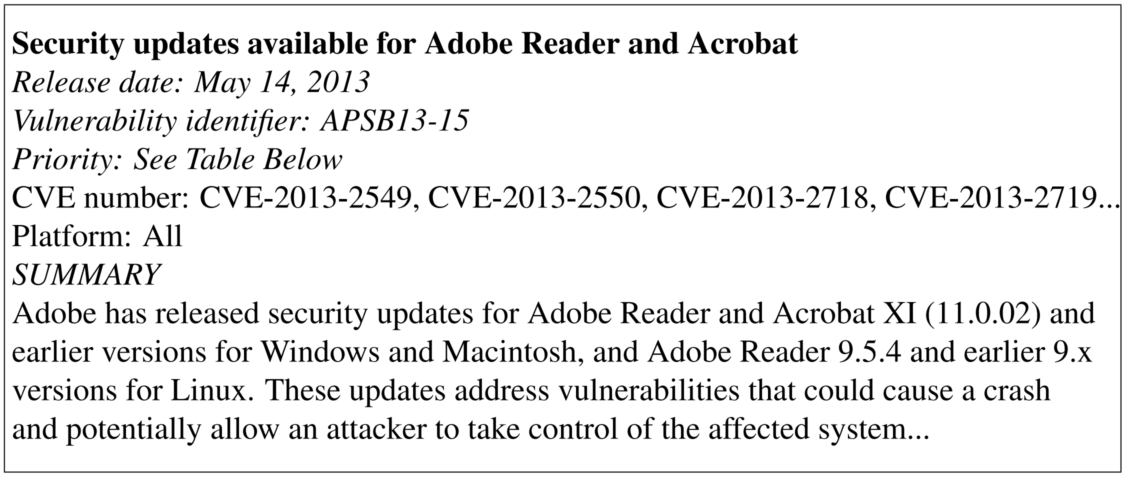


Microsoft Security Bulletin MS13-051 - Important
Vulnerability in Microsoft Office Could Allow Remote Code Execution (2839571)
Published: Tuesday, June 11, 2013
Version: 1.0
General Information
Executive Summary
This security update resolves one privately reported vulnerability in Microsoft Office. The vulnerability could allow remote code execution if a user opens a specially crafted Office document using an affected version of Microsoft Office software, or previews or opens a specially crafted email message in Outlook while using Microsoft Word as the email reader. An attacker who successfully exploited this vulnerability could gain the same user rights as the current user. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.
This security update...

FIG. 4.3. A sample excerpt of a Microsoft Security Bulletin

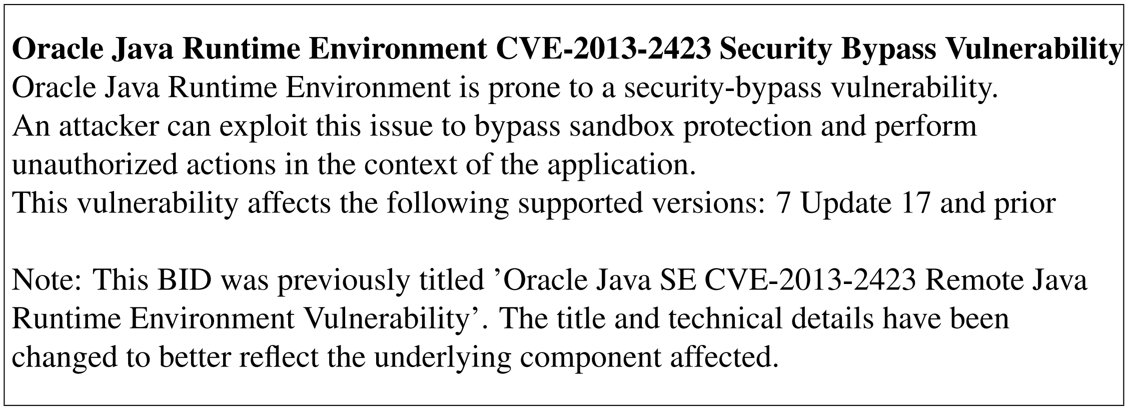
of a text from these documents. The style of documentation is mostly human-readable in plain text, though not sufficiently structured for machine interpretation.

We evaluated the performance of our concept extraction and linking framework for these non-CVE documents, via human annotation as described in the previous section. The extracted concepts and the relevant links were more in number compared to the annotations derived from the NVD/CVE text. When passed through DBpedia Spotlight, and provided with the same metrics for confidence and support parameters, a majority of the generated links were found to be inconsistent to the spotted concept. Figure 4.6 gives the precision, recall and F1 measurements of the concept extraction and linking framework on non-CVE data discusses above. As expected the precision measure decreased from 0.893 to 0.469. The recall measure was consistent and increased slightly from 0.63 to 0.69, while the F1 score reduced from 0.74 to 0.56. The change can be attributed to statistical variation, and the size limitation on the test set chosen for the non-NVD/CVE data.



Security updates available for Adobe Reader and Acrobat
Release date: May 14, 2013
Vulnerability identifier: APSB13-15
Priority: See Table Below
CVE number: CVE-2013-2549, CVE-2013-2550, CVE-2013-2718, CVE-2013-2719...
Platform: All
SUMMARY
Adobe has released security updates for Adobe Reader and Acrobat XI (11.0.02) and earlier versions for Windows and Macintosh, and Adobe Reader 9.5.4 and earlier 9.x versions for Linux. These updates address vulnerabilities that could cause a crash and potentially allow an attacker to take control of the affected system...

FIG. 4.4. A sample excerpt of an Adobe Security Bulletin



Oracle Java Runtime Environment CVE-2013-2423 Security Bypass Vulnerability
Oracle Java Runtime Environment is prone to a security-bypass vulnerability.
An attacker can exploit this issue to bypass sandbox protection and perform unauthorized actions in the context of the application.
This vulnerability affects the following supported versions: 7 Update 17 and prior

Note: This BID was previously titled ’Oracle Java SE CVE-2013-2423 Remote Java Runtime Environment Vulnerability’. The title and technical details have been changed to better reflect the underlying component affected.

FIG. 4.5. A sample excerpt of a SecurityFocus report

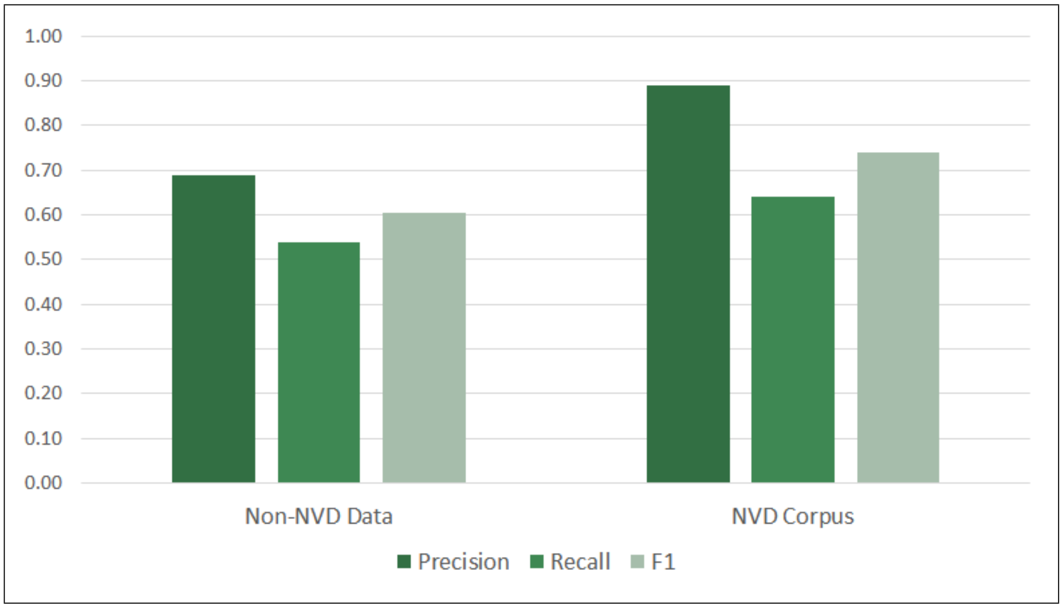


FIG. 4.6. Overall performance of the concept extraction and linking framework on non-NVD/CVE data

4.2 Challenges

The cybersecurity linked data generation framework provides a robust, scalable and effective model to identify terms and concepts described in the NVD dataset, gathers further information about the threat from associated repositories, extracts information from free text, models the data by aligning it to a custom ontology, and generates RDF linked data that is made available via a SPARQL endpoint.

However, this framework does have a few limitations, especially in terms of cross-referencing entities in the RDF graph based on the text description, and resolving identifies concepts to relevant documents/things on the Web.

4.2.1 Cross-Referencing Vulnerability Identifiers

The concept spotter and linking system does face certain challenges when identifying entities for some specific NVD descriptions that refer to other NVD CVE description.

There are certain sets of entries in the NVD repositories (mostly related to the same software product) that are observed to have the same summary description, with minor changes in the rest of the NVD (CVE, CVSS) properties. However, they provide references to other CVE IDs that might have the appropriate, more granular details regarding the attack.

Figure 4.7 shows an excerpt of the CVE-2013-0610 NVD entry that describes a buffer overflow attack on Adobe Acrobat and Reader. Although the NVD summary describes the means of the attack (*Buffer Overflow*) and the affected product (*Adobe Acrobat*), it does not provide further information such as the consequences. The description section mentions another CVE identifier and that this vulnerability is not different (similar) to CVE-2013-0626. Moreover, the severity score for the entry is 10 ("Critical"). There is a possibility that the descriptions and resource identified in the entry for CVE-2013-0626 might have more information about the vulnerability. We can gain such additional insight into the vulnerability description by cross-referencing of resources in the NVD RDF graph, based on mentions in text. The CVE ID mentioned in the current description can be extracted, mapped to the relevant NVD RDF entry in the graph and then cross-referenced with an appropriate object property. Retrieving the text associated with the referenced NVD CVE entries might help gather more information about the nature of such a critical attack, not only for a single CVE but a group of CVEs that might be reported together. In the future, we plan to consolidate these missed sources to give richer context on such vulnerabilities.

4.2.2 Unique identification of Vulnerability Concepts

Data integration using linked data technologies suggest the reusability of URIs that are resolvable and globally valid. As the desired outcome of a linked data effort is an integrated, well connected data corpus, associating concepts with referenceable resources from central sources (such as DBpedia) is highly recommended. However, not all concepts and terms spotted in the vulnerability descriptions can be associated with a valid, available resource.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ebqids: <http://ebiquity.umbc.edu/IDSv2.0.1.owl#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-0610>
ebqids:cveID "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-0610";
ebqids:cweID "http://cwe.mitre.org/data/definitions/119" ;
ebqids:summary "Stack-based buffer overflow in Adobe Reader and Acrobat 9.x
before 9.5.3, 10.x before 10.1.5, and 11.x before 11.0.1, not different
 from CVE-2013-0626." ;
ebqids:hasAccessComplexity "LOW" ;
ebqids:hasAccessVector "NETWORK" ;
ebqids:hasAuthentication "NONE" ;
ebqids:hasSeverityScore "10.0" ;
ebqids:hasVulnerabilitySource
"http://rhn.redhat.com/errata/RHSA-2013-0150.html" ,
"http://adobe.com/support/security/bulletins/apsb13-02.html" ,
"http://opensuse.org/opensuse-updates/2013-01/msg00081.html" ,
"http://opensuse.org/opensuse-security-announce/2013-01/msg00004.html" ,
"http://opensuse.org/opensuse-updates/2013-01/msg00028.html" ;
ebqids:hasMeans "dbpedia:Buffer_overflow" ;
ebqids:affectsProduct "dbpedia:Adobe_Acrobat" .
```

FIG. 4.7. An NVD entry excerpt which has an incomplete description, since it refers to another NVD CVE entry.

This may be the case when there is no relevant DBpedia resource available for the concept. As described in the Section 4.1, the number of concepts identified by the concept extractor framework is far larger than the available resources in the DBpedia knowledge base. The terms extracted by the cybersecurity entity and concept spotter, though not instantiated to relevant URIs, are important for profiling an attack.

This not only demonstrates the performance of our classifier, but also indicates the absence of entities that describe security concepts in the DBpedia knowledge base. In the future, we plan to resolve the unidentified concepts to external URIs that formally describe the security concept, and thereby reduce fact duplication and re-utilize existing URIs.

Chapter 5

CONCLUSION AND FUTURE WORK

In this work we introduced the Linked Data principles that define how to publish and interlink structured machine-readable data on the Web. We demonstrate a prototype for an entity and concept spotting framework that identifies cybersecurity-related concepts from heterogeneous data sources, aligns and links them to relevant resources on the Web using the IDS ontology, and generates an RDF linked data collection. The study promotes the concept of identifying every cybersecurity-related concept in terms of a dereferenceable and resolvable URI, and thereby rightfully justifies Google's Knowledge Graph's motto of "Things, not strings". We provide a semantic data representation for the concepts that are not limited to the NVD dataset. The linked data generation module leverages interoperability and reuse of URIs, thereby enhancing the binding with the Linked Open Data cloud.

There are several ways to improve the quality and the method of generating Linked Cybersecurity Data, in order to address the above mentioned issues of discoverability, sharing and reusability.

5.1 Knowledge Generation

As discussed in the previous chapter, it was observed that a substantial number of key security concepts, identified by the cybersecurity entity and concept extractor, either could not be linked to a relevant DBpedia resource or the document/description did not exist on the Web. In the former case, the usage of DBpedia Spotlight can be replaced by a dedicated and trained link generation framework. In the ideal case, as proposed by Mulwad et al. (4), we can use Wikitology (27) knowledge base together with the Wikipedia taxonomy for computer security exploits, in linking concepts from the Wikipedia knowledge base, to concepts identified through the concept extractor. However, Wikitology needs to be updated from its 2009 knowledge base to the current state for it to be useful.

There can be a case where no relevant description about a security concept is found in any Web-based document. In order to represent these terms in useful RDF instances, we plan to resolve the unidentified concepts to external URIs that formally describe the security concept, and thereby reduce fact duplication and re-utilize existing URIs. Hence our prototype can support knowledge generation of terms relevant to cybersecurity, that are unidentified or unknown, as relevant DBpedia resources.

On similar lines, the IDS ontology needs to be kept updated with the latest and greatest threat categories and instances of attacks. In cases where a new attack occurs which is not modeled in the IDS ontology, there is a need of an automated approach to dynamically add this threat as an instance of the appropriate attack class. There is also a need to address the problem of portability and dealing with ambiguity, where entities identified by our cybersecurity entity and concept spotter and previously not modeled in our IDS ontology can easily be mapped with high consistency. The class identification of annotated DBpedia resources, can be enhanced via measuring the score of textual similarity between results provided by the cybersecurity classifier and classes described in our vocabulary.

5.2 Adding non-NVD sources of information

Section 4.1.3 discussed the possibility of including other sources of information that do not have a structured representation, as is the case with the NVD schema. The concept extraction and linking component of our system was proved to provide promising results for extracting concepts and linking them to relevant DBpedia URIs, provided that the extractor is trained over a significant diverse set of cybersecurity-related documents such as SecurityFocus email alerts, Microsoft and Adobe security bulletins, Metasploit[1] reports, Nessus[2] scanner logs among others. By including such heterogeneous sources of data, we can prepare the system to identify concepts beyond the NVD schema. This will enable us to model much more information into the linked cybersecurity data cloud, most of which is described in unstructured pieces of text.

5.3 Make Linked Data more dynamic

Our current linked cybersecurity data generation framework is limited to pushing data from heterogeneous text sources into a structured and linked entity resource. However, the framework does not factor in changes to the current dataset provided by these text sources. This is true in case of the NVD RSS feeds. The entries provided by the RSS feeds are eight days worth of the latest NVD data, which is subject to change based on further analysis and incorporating information from other text sources. We plan to allow the framework to dynamically add or edit the concepts in the linked data. Our efforts will drive the linked data collection to the be a collaborative effort, and thereby move towards the idea of Read-Write Linked Data (28).

[1] http://www.metasploit.com/

[2] http://www.tenable.com/products/nessus

As the best practices suggest, we adopted existing deployed vocabularies such as the DBpedia ontology and built a custom cybersecurity vocabulary to model the concepts and entities identified in the NVD, CVE, CWE and CVSS data repositories. Our work focused on integrating heterogeneous data sources that identify properties of a security exploit, security concepts and threat descriptions, and represent them as RDF linked data. Building and maintaining such a rich source of cyber threat descriptions and metadata will not only increase the consumption of such data that was previously hidden in unstructured text, but also enhance discoverability, sharing and reusability of the data on the Semantic Web.

Our evaluation showed promising results for the concept extraction and linking framework. We plan to focus on further extracting previously unidentified security concepts from any given piece of text, identify properties and find relationships based on a heuristic. There are ongoing efforts to enhance the ontology to model detailed network-related terms and privacy concepts. The linked cybersecurity data graph can effectively support applications such as a search engine capable of making use of semantic technologies to model its knowledge base and to deliver content. We believe that expressing structured and unstructured cybersecurity-related text as linked data has potential to leverage automatic consumption and reasoning of security concepts, and can drive applications such as a situation aware intrusion detection system to detect and prevent potential “zero-day” attacks.

Appendix A

APPENDIX

```
<xsl:stylesheet version=”1.0”
xmlns:xsl=”http://www.w3.org/1999/XSL/Transform”
xmlns:cvss=”http://scap.nist.gov/schema/cvss-v2/0.2”
xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
xmlns:rdfs=”http://www.w3.org/2000/01/rdf-schema#”
xmlns:vuln=”http://scap.nist.gov/schema/vulnerability/0.4”
xmlns:ebqids=
”http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/v2.2/IDSOntology.owl/#”>
<xsl:output method=”xml” indent=”yes”\>
<xsl:template match=”/”>
<rdf:RDF>
<xsl:apply-templates\>
</rdf:RDF>
</xsl:template>
<xsl:variable name=”VulnerabilityURI”>
http://web.nvd.nist.gov/view/vuln/detail?vulnId=</xsl:variable>
<xsl:variable name=”WeaknessURI”>
http://cwe.mitre.org/data/definitions\</xsl:variable>
<xsl:variable name=”CVEURI”>
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=</xsl:variable>
<xsl:variable name=”cweID” select=”substring(vuln:cwe/@id,5)” \>
<xsl:template match =”/*/*”>
<xsl:variable name=”cveID” select=”vuln:cve-id” \> <xsl:element name=”rdf:Description”>
<xsl:attribute name=”rdf:about”>
<xsl:value-of select=”concat($VulnerabilityURI,$cveID)”\>
</xsl:attribute>
<ebqids:cveID>
<xsl:value-of select=”concat($CVEURI,$cveID)”\>
</ebqids:cveID>
<ebqids:hasPublishedDate>
<xsl:value-of select=”vuln:published-datetime”\>
</ebqids:hasPublishedDate>
<ebqids:hasModifiedDate>
<xsl:value-of select=”vuln:last-modified-datetime”\>
</ebqids:hasModifiedDate>
<ebqids:summary>
<xsl:value-of select=”vuln:summary”\>
</ebqids:summary>
<xsl:apply-templates select=”vuln:vulnerable-software-list”\>
<xsl:apply-templates select=”vuln:references”\>
<ebqids:hasSeverityScore>
<xsl:value-of select=”//cvss:score”\>
</ebqids:hasSeverityScore>
<ebqids:hasAccessVector>
<xsl:value-of select=”//cvss:access-vector”\>
</ebqids:hasAccessVector>
```

```
<ebqids:hasAccessComplexity>
<xsl:value-of select=”//cvss:access-complexity”\>
</ebqids:hasAccessComplexity>
<ebqids:hasAuthentication>
<xsl:value-of select=”//cvss:authentication”\>
</ebqids:hasAuthentication>
<ebqids:hasAvailabilityImpact>
<xsl:value-of select=”//cvss:availability-impact”\>
</ebqids:hasAvailabilityImpact>
<ebqids:hasIntegrityImpact>
<xsl:value-of select=”//cvss:integrity-impact”\>
</ebqids:hasIntegrityImpact>
<ebqids:hasConfidentialityImpact>
<xsl:value-of select=”//cvss:confidentiality-impact”\>
</ebqids:hasConfidentialityImpact>
<ebqids:hasSource>
<xsl:value-of select=”//cvss:source”\>
</ebqids:hasSource>
<xsl:apply-templates select=”vuln:cwe/@id”>
<xsl:with-param name=”cweID”
select=”substring(vuln:cwe/@id,5)”\>
</xsl:apply-templates>
</xsl:element>
</xsl:template>
<xsl:template match=”vuln:vulnerable-software-list”>
<xsl:for-each select=”vuln:product”>
<ebqids:affectsProduct>
<xsl:value-of select=”.”\>
</ebqids:affectsProduct>
</xsl:for-each>
</xsl:template>
<xsl:template match=”vuln:product”>
<ebqids:product> </ebqids:product>
</xsl:template>
<xsl:template match=”vuln:references”>
<ebqids:hasVulnerabilitySource>
<xsl:value-of select=”vuln:reference/@href”\>
</ebqids:hasVulnerabilitySource>
</xsl:template>
<xsl:template match=”vuln:cwe/@id”>
<xsl:param name=”cweID” \>
<ebqids:cweID>
<xsl:value-of select=”concat($WeaknessURI,$cweID)”\>
</ebqids:cweID>
</xsl:template>
</xsl:stylesheet>
```

FIG. A.1. The XSL transformation for converting NVD XML tags to RDF triples

```
<?xml version=“1.0” encoding=“UTF-8”?>
<nvd xmlns:vuln=“http://scap.nist.gov/schema/vulnerability/0.4”
xmlns:cvss=“http://scap.nist.gov/schema/cvss-v2/0.2”>
<entry id=“CVE-2012-0150”>
<vuln:vulnerable-software-list>
<vuln:product>cpe:/o:microsoft:windows_vista::sp2:x64 </vuln:product>
<vuln:product>cpe:/o:microsoft:windows_7:::x86 </vuln:product>
<vuln:product>cpe:/o:microsoft:windows_7::sp1:x86 </vuln:product>
<vuln:product>cpe:/o:microsoft:windows_vista::sp2 </vuln:product>
</vuln:vulnerable-software-list>
<vuln:cve-id>CVE-2012-0150</vuln:cve-id>
<vuln:cvss>
<cvss:base_metrics>
<cvss:score>9.3</cvss:score>
<cvss:access-vector>NETWORK</cvss:access-vector>
<cvss:access-complexity>MEDIUM</cvss:access-complexity>
<cvss:authentication>NONE</cvss:authentication>
</cvss:base_metrics>
</vuln:cvss>
<vuln:cwe id=“CWE-119” />
<vuln:references xml:lang=“en” reference_type=”VENDOR_ADVISORY”>
<vuln:source>MS</vuln:source>
<vuln:reference href=“http://technet.microsoft.com/security/bulletin/MS12-013”
xml:lang=“en”>MS12-013</vuln:reference>
</vuln:references>
<vuln:summary>Buffer overflow in msvcrt.dll in Microsoft Windows Vista SP2,
Windows Server 2008 SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 allows
remote attackers to execute arbitrary code via a crafted media file,
aka “Msvcrt.dll Buffer Overflow Vulnerability.”
</vuln:summary>
</entry>
</nvd>
```

FIG. A.2. An excerpt of an NVD XML entry

```
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ebqids:<http://ebiquity.umbc.edu/IDSv2.0.1.owl#> .
@prefix dbpedia:<http://dbpedia.org/resource/> .
<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2012-0150>
ebqids:cveID “http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0150” ;
ebqids:cweID “http://cwe.mitre.org/data/definitions/119” ;
ebqids:affectsProduct “dbpedia:Windows_Vista” , ”dbpedia:Windows_7” ;
ebqids:summary “Buffer overflow in msvcrt.dll in Microsoft Windows Vista SP2, Windows
Server 2008 SP2, R2, and R2 SP1, and Windows 7 Gold and SP1 allows remote attackers
to execute arbitrary code via a crafted media file,
aka “Msvcrt.dll Buffer Overflow Vulnerability.”” ;
ebqids:hasAccessComplexity “MEDIUM” ;
ebqids:hasAccessVector “NETWORK” ;
ebqids:hasAuthentication “NONE” ;
ebqids:hasSeverityScore “9.3” ;
ebqids:hasVulnerabilitySource
“http://technet.microsoft.com/security/bulletin/MS12-013” ;
ebqids:hasMeans “dbpedia:Buffer_overflow” ;
ebqids:hasConsequence “dbpedia:Arbitrary_code_execution” ;
ebqids:hasTerms “http://dbpedia.org/resource/Computer_file” ,
“http://dbpedia.org/resource/Dynamic-link_library” ,
“http://dbpedia.org/resource/Vulnerability_(computing)” .
```

FIG. A.3. Turtle representation of extracted information

REFERENCES

[1] Cyber criminals target Skype, Facebook and Windows users. http://bit.ly/cyberCriminals.

[2] Stephen D. Quinn, David A. Waltermire, Christopher S. Johnson, Karen A. Scarfone, and John F. Banghart. SP 800-126. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.0. Technical report, National Institute of Standards & Technology, Gaithersburg, MD, United States, 2009.

[3] S. More, M. Matthews, A. Joshi, and T. Finin. A Knowledge-Based Approach to Intrusion Detection Modeling. In *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, pages 75–81, 2012.

[4] V. Mulwad, Wenjia Li, A. Joshi, T. Finin, and K. Viswanathan. Extracting Information about Security Vulnerabilities from Web Text. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 3, pages 257–260, 2011.

[5] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[6] T Reuters. OpenCalais, 2009.

[7] Giuseppe Rizzo and Raphaël Troncy. NERD: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstra-*

tions at the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL '12, pages 73–76, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[8] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[9] Ravendar Lal. Information Extraction of Security related entities and concepts from unstructured text. Master's thesis, Universityb of Maryland Baltimore County, 2013.

[10] Vaibhav Khadilkar, Jyothsna Rachapalli, and Bhavani Thuraisingham. Semantic Web Implementation Scheme for National Vulnerability Database (Common Platform Enumeration Data). Technical Report UTDCS-01-10, University of Texas at Dallas, 2010.

[11] Jeffrey Undercoffer, John Pinkston, Anupam Joshi, and Timothy Finin. A Target-Centric Ontology for Intrusion Detection. In *Proceeding of the IJCAI-03 Workshop on Ontologies and Distributed Systems*, pages 47–58. Morgan Kaufmann, 2004.

[12] Tim Berners-Lee. Design issues: Linked data. http://www.w3.org/DesignIssues/LinkedData.html, 2006.

[13] Linked Data. http://linkeddata.org/, 2007.

[14] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[15] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media Meets Semantic Web How the BBC Uses DBpedia and Linked Data to Make Connections. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvnen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 723–737. Springer Berlin Heidelberg, 2009.

[16] Bernadette Hyland, Ghislain Atemezing, Michael Pendleton, and Biplav Srivastava. Linked Data Glossary - W3C Working Group Note, June 2013.

[17] Protege Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu.

[18] B Motik, U Sattler, M Smith, PF Patel-Schneider, B Parsia, C Bock, A Fokoue, P Haase, R Hoekstra, I Horrocks, et al. OWL 2 Web Ontology Language: structural specification and functional-style syntax. 2008.

[19] Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.

[20] Tom Heath and Christian Bizer. Linked Data: Evolving the Web into a Global Data Space. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1(1):1–136, February 2011.

[21] Diego Berrueta, Sergio Fernndez, and Ivn Frade. Cooking HTTP content negotiation with Vapour. In *In Proceedings of 4th workshop on Scripting for the Semantic Web 2008 (SFSW2008). co-located with ESWC2008*, 2008.

[22] Rachel Wadham. Rich site summary (RSS). *Library Mosaics*, 16(1):25, 2005.

[23] Richard Cyganiak and Chris Bizer. Pubby - A Linked Data Frontend for SPARQL Endpoints. http://wifo5-03.informatik.uni-mannheim.de/pubby/, 06 2007.

[24] Annika Flemming. Quality Criteria for Linked Data sources. Master's thesis, University of Berlin, Humboldt, 2011.

[25] Giuseppe Rizzo and Raphaël Troncy. NERD: evaluating named entity recognition tools in the web of data. In *ISWC 2011, Workshop on Web Scale Knowledge Extraction (WEKEX'11), October 23-27, 2011, Bonn, Germany*, Bonn, GERMANY, 10 2011.

[26] Precision and recall. `http://en.wikipedia.org/wiki/Precision_and_recall`.

[27] Zareen Saba Syed. *Wikitology: a novel hybrid knowledge base derived from Wikipedia*. PhD thesis, University of Maryland Baltimore County, Catonsville, MD, USA, 2010. AAI3422868.

[28] Tim Berners-Lee and Kieron O'Hara. The read-write linked data web. *Philosophical Transactions of the Royal Society A: Mathematical Physical and Engineering Sciences*, 371(1987):20120513–[5pp], March 2013.